

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Analýza pohybu osob založená na zpracování obrazu

Diplomová práce

Vedoucí práce:
Ing. Vít Ondroušek Ph.D.

Bc. Robin Antonič

Brno 2017

Rád bych poděkoval mému vedoucímu Ing. Vítu Ondrouškovi, Ph. D. za jeho odborné vedení, rady a pomoc při psaní této práce.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Analýza pohybu osob založená na zpracování obrazu**

vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 19.května 2017

.....

Abstract

Bc. Antonič, R. Analysis of a pedestrian trajectories based on Image Processing. Diploma thesis. Brno 2017.

This diploma thesis is focused on design of methodology for detecting and analyzing pedestrian trajectories, moving in a predefined area, which is recorded by surveillance cameras. Design approach is mainly based on optical flow algorithms. The designed methodology was implemented into a complex software by using framework .NET. The developed application is able to display elementary time-space characteristics using a various tools, which provide sophisticated view on the detected pedestrian trajectories in a predefined area. The designed methodology was verified by performing two complex various tests.

Abstrakt

Bc. Antonič, R. Analýza pohybu osob založená na zpracování obrazu. Diplomová práce. Brno 2017.

Práce se zabývá návrhem metodiky pro detekci a analýzu trajektorií osob pohybujících se ve vytyčeném prostoru, který je snímán průmyslovými kamerami. Navržený přístup využívá vybrané algoritmy pro výpočet optického toku. Navržená metodika byla následně implementována do komplexního software pomocí frameworku .NET. Aplikace umožňuje prezentovat základní časoprostorové charakteristiky dle různých nástrojů, které poskytují sofistikovanější pohled na pohyb osob ve snímané oblasti. Dále byla provedena verifikace navržené metodiky na základě dvou komplexních testů.

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod a cíl práce | 11 |
| 1.1 | Úvod práce | 11 |
| 1.2 | Cíl práce | 11 |
| 2 | Teoretická část | 12 |
| 2.1 | Metody pro detekci pohybu | 12 |
| 2.2 | Přířazovací problém | 19 |
| 2.3 | Kalmanův filtr | 21 |
| 2.4 | Zpracování obrazu | 23 |
| 2.5 | Znamé detektory osob | 32 |
| 2.6 | Charakteristiky vyhodnocení | 33 |
| 2.7 | Vybrané nástroje | 35 |
| 3 | Vstupní data analyzovaného problému | 37 |
| 3.1 | Popis pořízených videí | 37 |
| 3.2 | Popis datových sad | 37 |
| 4 | Komparace vybraných metod | 41 |
| 4.1 | Komparace metod pro výpočet optického toku | 41 |
| 4.2 | Komparace metod pro klasifikaci obrazu | 43 |
| 5 | Návrh metodiky vyhodnocení pohybu osob | 47 |
| 5.1 | Metodika detekce trajektorií | 47 |
| 5.2 | Metodika analýzy trajektorií | 51 |
| 6 | Tvorba řídicí aplikace | 54 |
| 6.1 | Uživatelský pohled | 54 |
| 6.2 | Programátorský pohled | 61 |
| 7 | Dosažené výsledky | 65 |
| 7.1 | Testování detekce pohybu osob v prostoru | 65 |
| 7.2 | Detekce příchodů a odchodů z místnosti | 67 |
| 8 | Zhodnocení | 70 |
| 8.1 | Diskuze | 70 |
| 8.2 | Shrnutí | 71 |
| 8.3 | Závěr | 72 |
| 9 | Literatura | 73 |
| | Přílohy | 76 |
| A | Diagram tříd jmenného prostoru PedestrianVerification | 77 |

| | | |
|----------|---|-----------|
| B | Diagram tříd jmenného prostoru MotionDetection | 78 |
| C | Diagram tříd jmenného prostoru ObjectsIdentification | 79 |
| D | Grafické rozhraní pro nastavení algoritmu | 80 |

1 Úvod a cíl práce

1.1 Úvod práce

Počítačové vidění je stále více využívané odvětví výpočetní techniky, které je zaměřeno na získávání informací ze snímaného obrazu. V dnešní době, kdy stále roste počet kamer napojených na centrální systémy, je tedy možné využít oblast počítačového vidění k analyzování chování pohybujících se osob ve snímaném prostoru. Analýza chování pohybujících se osob v prostoru může být, například použita pro analýzu nákupního chování zákazníka nebo pro podporu manažerského rozhodování.

Diplomová práce je zaměřena na návrh metodiky pro detekci a analýzu trajektorií pohybu osob a její následnou implementaci do komplexního software. Práce je také zaměřena na reprezentaci získaných informací o pohybu osob pomocí nástrojů, které dokáží vhodným způsobem zobrazit nalezené informace a poskytnout další pohledy na chování osob ve snímané oblasti.

1.2 Cíl práce

Práce je zaměřena na problematiku vyhodnocování chování dynamických objektů z vytyčeného prostoru (např. prostory obchodních center, obchodů, otevřených kancelářských místností apod.), který je snímán průmyslovými kamerami, na základě analýzy obrazu. Dílčím cílem práce je návrh metodiky pro detekci trajektorií pohybu osob na základě vybraných algoritmů pro výpočet optického toku a také návrh metodiky pro analýzu detekovaných trajektorií pomocí vybraných klasifikačních algoritmů a známých detektorů osob v obraze. Souvisejícími cíli je tvorba komplexního software, který implementuje navrženou metodiku a verifikace navržené metodiky pomocí komplexních testů.

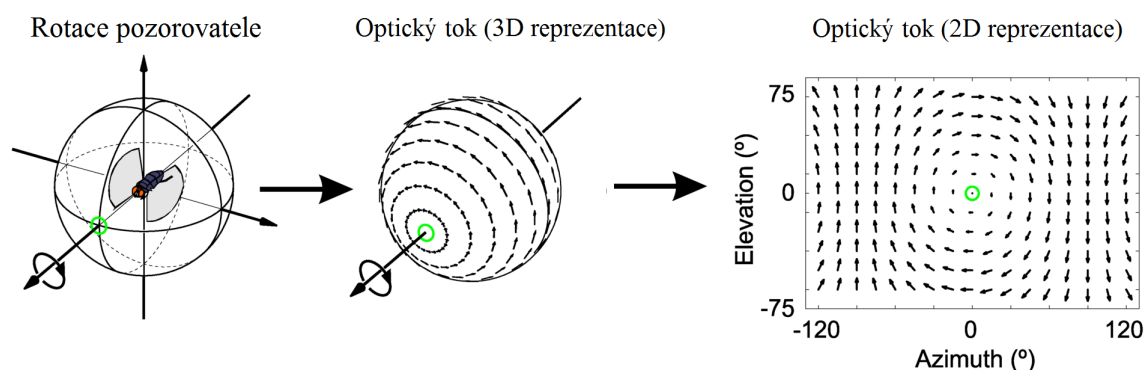
2 Teoretická část

2.1 Metody pro detekci pohybu

Tato sekce je zaměřena na metody, které dokáží odhadovat dynamické objekty respektive jejich intenzitu pohybu z dynamického obrazu. Dynamický obraz reprezentuje sledovanou scénu a je složen ze sekvence statických obrazů, které jsou uspořádány v čase. V této práci jsou využívány následující metody:

Optický tok

Optický tok (angl. Optical flow) zachycuje rozložení zdánlivého pohybu objektů, které vzniklo mezi dvěma snímky dynamického obrazu, pomocí vektorového pole. Optický tok je tedy zdánlivý pohyb pozorované scény, který může být pozorován pohybujícím se pozorovatelem scény, viz Obrázek č. 1.



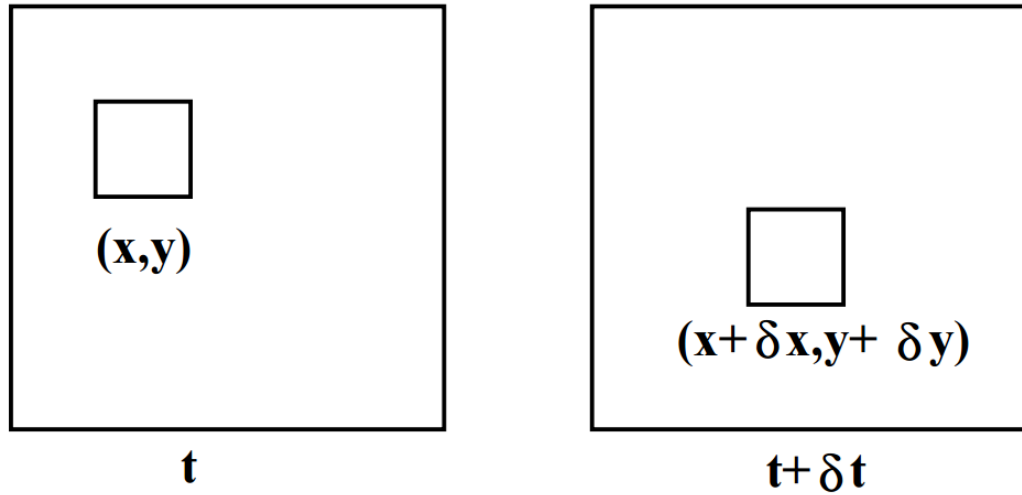
Obrázek 1: Pozorovatel a jeho vliv na optický tok změnou pohybu. Upraveno ze zdroje: (Huston, Krapp, 2008).

Optický tok je reprezentován vektorovým polem, které každému bodu v obraze přiřazuje vektor s určitou intenzitou a směrem vektoru vzhledem k předchozímu snímku. Vektory tedy uchovávají informace o vyjadřující rychlosti a směru pohybu pro každý bod v obraze. Obecně platí, že objekty, které se nacházejí blíže k pozorovateli, vykazují vyšší rychlost pohybu a objekty, které jsou vzdáleny, naopak vykazují menší pohyb ve scéně.

Pro výpočet optického toku jsou využívány dva různé přístupy. První přístup pracuje se všemi body v obraze. Tento přístup se nazývá hustý optický tok (angl. Dense optical flow). Druhým přístupem pro nalezení optického toku je řídký optický tok (angl. Sparse optical flow), který využívá pouze vybranou množinu jednoduše detekovatelných bodů v obraze, což vede ke snížení časové náročnosti výpočtu (Orsák, 2014).

Princip optického toku

Funkce $I(x, y, t)$ udává hodnotu intenzity pro bod ze souřadnic (x, y) z okolí všech pixelů $n \cdot n$ v čase t . Pohyb tohoto bodu ve stejném obrazu můžeme popsat pomocí funkce $I(x + dx, y + dy, t + dt)$, viz Obrázek č. 2.



Obrázek 2: Zachycení změny polohy vybraného bodu v dynamickém obrazu. Upraveno ze zdroje: (Barron, Thacker, 2005).

Lze tedy sestavit následující rovnici:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \quad (1)$$

rozšířením této rovnice pomocí Taylorova rozvoje dostaneme:

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{dI}{dx} \cdot dx + \frac{dI}{dy} \cdot dy + \frac{dI}{dt} \cdot dt + \varepsilon \quad (2)$$

Kde ε udává členy vyššího řádu Taylorova rozvoje. Zavedeme substituci $v_x = \frac{dx}{dt}$, $v_y = \frac{dy}{dt}$. Po úpravách rovnice dostáváme diferenciální rovnici ve tvaru:

$$\frac{dI}{dx} \cdot v_x + \frac{dI}{dy} \cdot v_y + \frac{dI}{dt} = 0 \quad (3)$$

Parciální derivace ve směru x , y a v čase t zapíšeme ve tvaru $I_x = \frac{dI}{dx}$, $I_y = \frac{dI}{dy}$, $I_t = \frac{dI}{dt}$ a dosadíme do rovnice, čímž dostaneme tvar:

$$(I_x, I_y) \cdot (v_x, v_y) = -I_t \quad (4)$$

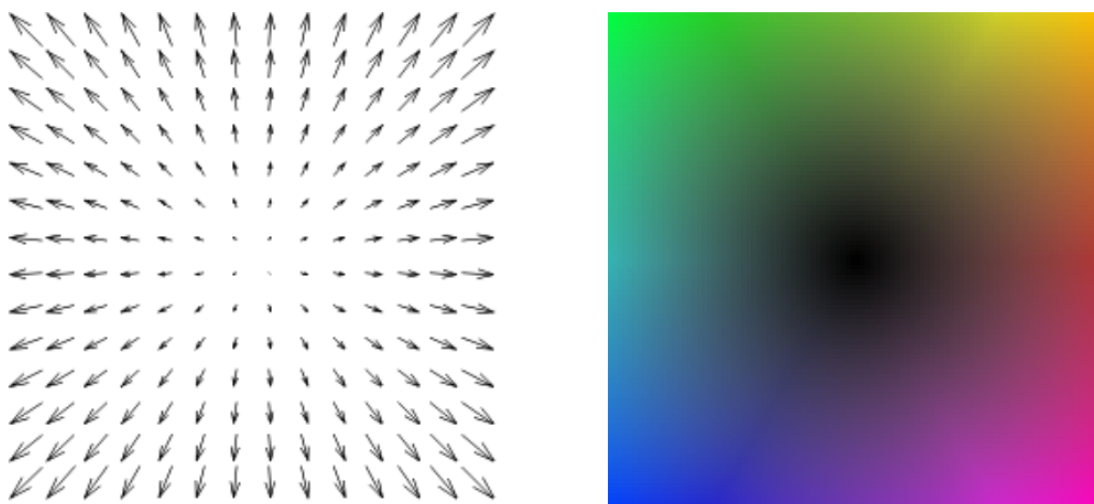
Tuto rovnici převede na finální tvar, který je nazýván jako rovnice omezení optického toku:

$$\nabla I \cdot \vec{v} = -I_t, \quad (5)$$

Vektor $\vec{v} = (v_x, v_y)$ udává optický tok pro bod ze souřadnic (x, y) v čase t . Gradient $\nabla I = (I_x, I_y)$ je složen z parciálních derivací I_x, I_y ve směru x, y . Rovnice omezení optického toku obsahuje dvě neznámé tj. (v_x, v_y) pro každý pixel, což je důsledek vzniku problému apertury. Pro vyřešení tohoto problému a tedy pro nalezení optického toku musí být zavedeny další omezení (Barron, Thacker, 2005).

Reprezentace optického toku

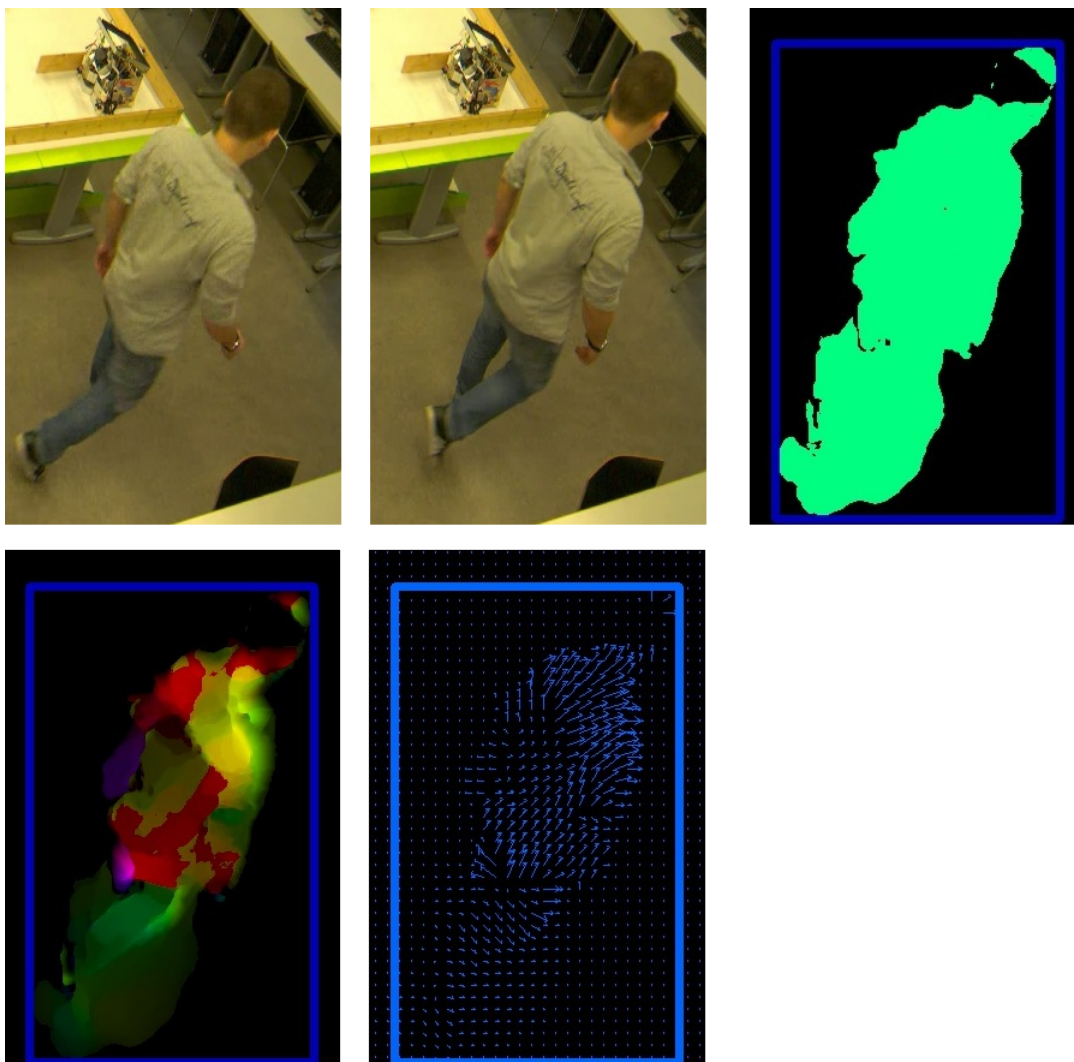
Výchozí reprezentace optického toku je vektorové pole, které obsahuje výše zmíněné vektory, udávající intenzitu a směr pohybu pro každý bod scény. Jednotlivé směrové vektory lze převést do modelu HSV podle následujícího schématu, který je znázorněn v obrázku č. 3.



Obrázek 3: Schéma pro převedení vektorového pole na model HSV. Zdroj: (Sánchez, Meinhardt-Llopis, Facciolo, 2013).

Následně je každý vektor reprezentován v tomto modelu dle své délky, která je určena složkou S neboli sytostí barvy a svým směrem tedy složkou H, tj. barevným tónem. Výsledný převod optického toku do HSV modelu poskytuje další možnosti, jakým způsobem lze zpracovat pohyb v dynamickém obraze.

Jednotlivé způsoby reprezentace optického toku jsou zobrazeny dle série snímků v obrázku č. 4. Obrázek zobrazuje fragmenty pohybu člověka mezi dvěma po sobě jdoucími snímky v laboratoři Q11. Následující fragment zobrazuje optický tok v jedné barvě, což vede ke ztrátě informace o rychlosti a směru dynamického objektu, ale fragment může být následně použit pro segmentaci obrazu. V dalším fragmentu je zachycena reprezentace dle modelu HSV, kde je patrné, že jednotlivé části těla nevykazují stejný směr pohybu, i když celý objekt se pohybuje v jednom směru.



Obrázek 4: Série snímků, které zachycují pohyb osoby a různé zobrazení optického toku.

V posledním fragmentu je zobrazeno vektorové pole, které obsahuje dílčí vektory pohybu. Z vektorového pole je patrné, které části těla vykazovali jiný směr a intenzitu pohybu než kompletní celek. Vektorové pole tedy může dle vektorů v daném objektu, poskytnout informace o směru a rychlosti celého objektu mezi snímky.

Vybrané metody založené na optickém toku

V této části budou popsány vybrané metody pro výpočet optického toku, které byly použity v praktické části práce. Metody jsou rozděleny následovně:

Färnebackova metoda

Algoritmus, který slouží pro odhad pohybu mezi dvěma snímky. Algoritmus používá kvadratické polynomy k přibližnému odhadu pohybu mezi snímky s využitím transformace polynomiálního rozvoje. Polynomiální rozvoj tedy přibližně odhaduje okolí každého pixelu obrazu pomocí kvadratického polynomu. Kvadratické polynomy vytvářejí lokální model signálu, který je vyjádřen v lokálním souřadném systému.

$$f(x) \sim x^T \cdot A \cdot x + b^T \cdot x_c \quad (6)$$

Přičemž struktura A je symetrická matice, b vektor a c skalár. (De Boer, Kalksma, 2015)

Färnebackův algoritmus využívá pro výpočet optického toku všechny pixely obrazu, proto je výpočet pomalejší oproti metodám, které využívají pouze vybrané pixely např. metoda Lucas-Kanade. Výsledkem algoritmu je vektorové pole obsahující směrové vektory.

Lucas - Kanade metoda

Metoda využívá lokální omezení pohybu k vyřešení problému apertury. Tato metoda předpokládá, že malé oblasti v obraze odpovídají stejnému objektu a mají podobný pohyb. Celý obraz je rozdělen na dílčí okna o velikosti $N \cdot N$, kdy každé okno obsahuje $p = N^2$ pixelů. Pomocí lokálního omezení pohybu je vytvořen systém, který obsahuje p rovnic a lze jej zapsat:

$$I_{x1} \cdot u + I_{y1} \cdot v + I_{t1} = 0 \quad (7)$$

$$I_{x2} \cdot u + I_{y2} \cdot v + I_{t2} = 0 \quad (8)$$

$$\cdot \quad (9)$$

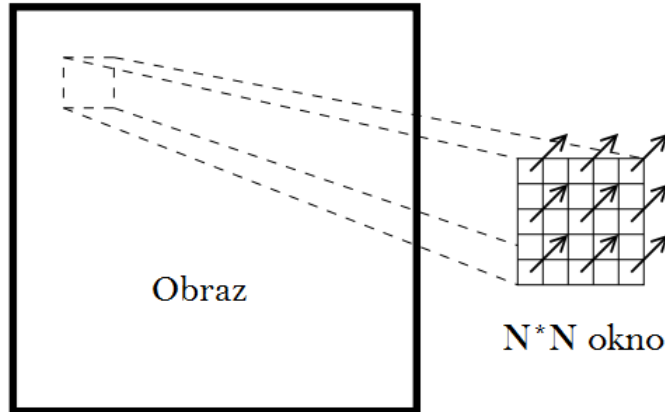
$$\cdot \quad (10)$$

$$\cdot \quad (11)$$

$$I_{xp} \cdot u + I_{yp} \cdot v + I_{tp} = 0 \quad (12)$$

Tento systém rovnic lze vyřešit pomocí metody LMS, což povede k výpočtu odhadu vektoru optického toku. Odhadovaný optický tok pro každé okno odpovídá vektoru optického toku pro všechny pixely v daném okně (De Boer, Kalksma, 2015).

Základní myšlenka algoritmu je založena na těchto předpokladech (Kaehler, Bradski, 2008):



Obrázek 5: Vektor optického toku pro všechny pixely v daném oknu. Upraveno ze zdroje: (De Boer, Kalksma, 2015)

1. **Konstantní jas** - Jas pixelů objektu ve scéně se nemění mezi snímky, takovým způsobem, že by se změnil vzhled objektu. Předpoklad lze matematicky zapsat následovně:

$$f(x, t) I(x(t), t) = I(x(t + dt), t + dt) \quad (13)$$

2. **Dočasná persistence** - Dočasné přírůstky pohybu jsou dostatečně rychlé vzhledem k míře pohybu v obraze takovým způsobem, že objekt ve scéně se nepohybuje moc rychle mezi snímky scény.
3. **Prostorová koherence** - Sousední body pixelů ve scéně patří do stejné roviny a mají podobný pohyb.

TV- L^1 metoda

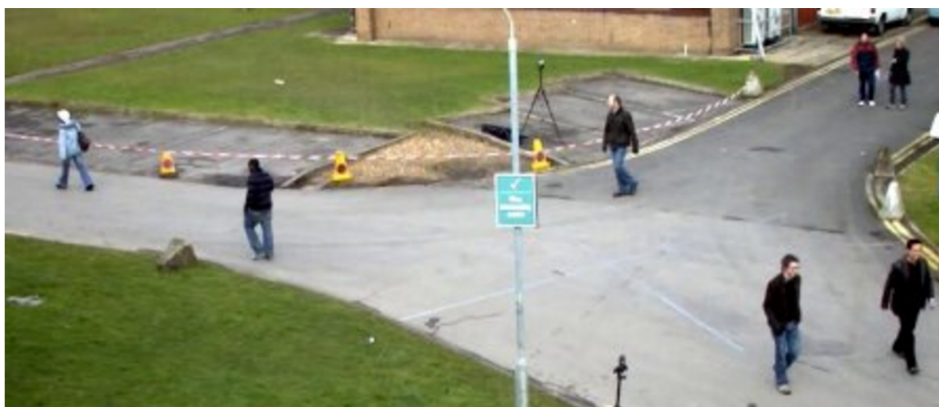
Metoda patří mezi variační metody, které patří mezi nejúspěšnější metody pro výpočet optického toku mezi dvěma snímky. V základním nastavení se zpracovávají dva snímky I_0 a $I_1 : (\Omega \subset \mathbb{R}^2) \rightarrow \mathbb{R}$. Cílem je nalézt mapu disparity $u : \Omega \rightarrow \mathbb{R}^2$, která minimalizuje obrazové chybné kritérium s regulační silou. Algoritmus využívá rozdíl intenzity mezi pixely jako skóre pro podobnost obrazů. Cílová mapa disparity u je minimalizována:

$$\int_{\Omega} \{\lambda \phi(I_0(x) - I_1(x + u(x))) + \psi(u, \Delta u \dots)\} dx \quad (14)$$

Kde $\phi(I_0(x) - I_1(x + u(x)))$ udává výstižnost obrazových dat a $\psi(u, \Delta u \dots)$ vyjadřuje regularizační podmínku. λ popisuje váhy mezi regulační silou a výstižností dat (Wedel, Pock, Zach, Bischof, 2009).

Princip rozdílové metody

Odečet pozadí (angl. Background Subtractor) je široce využívaný přístup pro detekci dynamických objektů z obrazu pořízeného statickými kamerami. Podstata těchto metod je založena na rozdílu požadovaného a referenčního snímku. Referenční snímek je někdy také nazýván model pozadí. Referenční snímek musí zachycovat požadovanou scénu bez dynamických objektů a také musí být pravidelně aktualizován, aby se adaptoval na variabilní světelné podmínky a nastavení geometrie (Piccardi, 2004).



Obrázek 6: Vstupní obraz pro rozdílovou metodu. Zdroj: (OpenCV 3.0, 2014)

Rozdílová metoda MOG

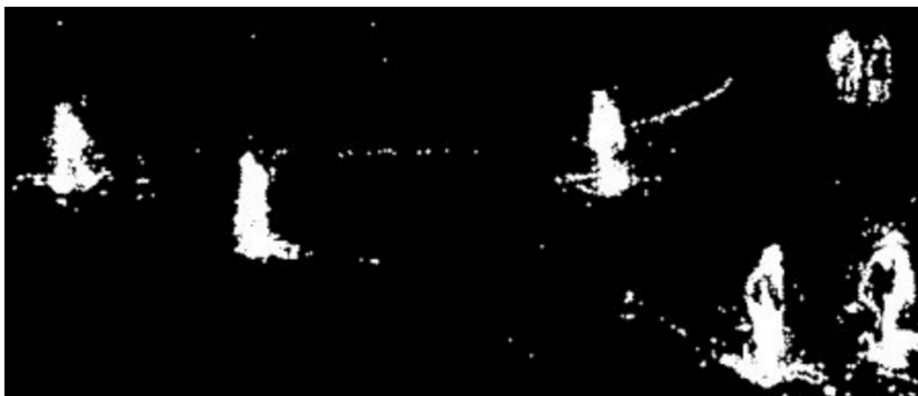
Tato metoda umožňuje segmentaci pozadí či popředí pomocí směsice Gaussovských hustot pravděpodobnosti. Metoda byla navržena v roce 2001, využívá algoritmy pro modelování každého pixelu v pozadí pomocí směsice K Gaussovských rozdělání (OpenCV 3.0, 2014).



Obrázek 7: Výstup rozdílové metody MOG. Zdroj: (OpenCV 3.0, 2014)

Rozdílová metoda GMG

Tato metoda byla publikována v roce 2012. Její princip vychází z kombinace statistického odhadu pozadí obrazu a Bayesovské segmentace pro každý pixel. Ve výchozím nastavení používá metoda prvních 120 snímků pro modelování pozadí. Využívá také algoritmus pro pravděpodobnostní segmentaci popředí, který vychází z Bayesova teorému pro identifikaci možných objektů v popředí. Morfologické operace typu otevření, uzavření a dalších jsou použity k odstranění nechtěného šumu. Pro prvních pár snímků dostaneme aplikováním této metody na vstupní snímek pouze černý obraz (OpenCV 3.0, 2014).



Obrázek 8: Výstup rozdílové metody GMG. Zdroj: (OpenCV 3.0, 2014)

2.2 Přiřazovací problém

Podstatou tohoto problému je vhodné přiřazení m zdrojů k m cílům takových způsobem, aby se dosáhlo optimálního přiřazení. Zdroje jsou přiřazeny k cílům podle následujících pravidel (Friebeľová, 2006):

1. Každý zdroj musí být přiřazen k cíli
2. Ke každému cíli musí být přiřazen zdroj
3. Celkové přiřazení musí být optimální

$$\sum_{j=1}^m x_{ij} = 1, \quad i = 1, 2, \dots, m \quad (15)$$

$$\sum_{i=1}^m x_{ij} = 1, \quad j = 1, 2, \dots, m \quad (16)$$

$$\sum_{i=1}^m \sum_{j=1}^m x_{ij} \cdot c_{ij} \rightarrow \text{extrém} \quad (17)$$

Sazby c_{ij} určují hodnotu přiřazení např. čas, který i -tý pracovník potřeboval pro výrobu produktu na j -tém stroji. Proměnné x_{ij} slouží jako identifikátory přiřazení a mohou nabývat hodnoty 0 či 1 na základě toho, jestli je i -tý zdroj přiřazen k j -tému cíli (Friebeľová, 2006).

Maďarská metoda

Tato metoda využívá tabulku, jejíž řádky odpovídají zdrojům a sloupce cílovým objektům. V jednotlivých políčkách tabulky jsou uvedeny jednotlivé sazby c_{ij} . Dále musí být také splněna podmínka řešitelnosti, tedy musí odpovídat počet zdrojů počtu cílů, pokud toto není splněno, tak je zaveden fiktivní zdroj či cíl (Friebeľová, 2006).

| | C_1 | C_2 | C_3 | |
|-------|-------|-------|-------|---|
| Z_1 | 2 | 5 | 6 | 1 |
| Z_2 | 3 | 8 | 4 | 1 |
| Z_3 | 4 | 9 | 7 | 1 |
| | 1 | 1 | 1 | |

Obrázek 9: Výchozí tabulka maďarské metody. Zdroj: (Friebeľová, 2006)

Sazby jsou následně zapsány do matice, ve které jsou provedeny následující úkony:

1. Redukce řádků
2. Redukce sloupců
3. Výběr nezávislých nul
4. výpočet účelové funkce

5. Ověření výběru nezávislých nul pomocí Königovy věty
6. Výběr řady neobsahující nezávislé nuly a nulové prvky
7. Redukování sazeb
8. Nalezení nezávislých nul v nově vypočtené matici - Postup se opakuje od bodu 3), dokud není nalezeno m nezávislých nul

2.3 Kalmanův filtr

Hlavní pozornost v oblasti teorii řízení si v 60 letech získalo využívání vnitřního stavového popisu. Úlohy filtrace, interpolace a predikce, které jsou formulovány pomocí vnitřního popisu systému lze řešit, pokud je možné odhadovat stav systému pomocí pozorování vstupů a výstupů systému. Pokud je daný problém formulován deterministicky, tak lze sestavit pozorovatele stavu, který slouží pro odhad stavu. Pokud je problém formulován stochasticky, tak lze formulovat úlohu odhadu dle optimálního LMS odhadu stavu a výsledný optimální pozorovatel je nazýván Kalmanův filtr (Havlena, Štecha, 1999).

Nechť máme následující lineární stochastický systém:

$$x(t+1) = Ax(t) + Bu(t) + v(t) \quad (18)$$

$$y(t) = Cx(t) + Du(t) + e(t) \quad (19)$$

Přičemž platí, že A , B , C a D jsou známé matice a proměnné $v(t)$ a $e(t)$ jsou nekorelované diskrétní bílé šумы. Matice kovariance je formulována následovně:

$$\varepsilon \left\{ \begin{bmatrix} v(t) \\ e(t) \end{bmatrix} \cdot \begin{bmatrix} v(t) \\ e(t) \end{bmatrix}^T \right\} = \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \quad (20)$$

Cílem je nalezení algoritmu, který je schopen generovat matice kovariance chyb odhadu a posloupnost lineárních odhadů stavu (t) .

$$P(t) = \varepsilon \left\{ (x(t) - \hat{x}(t))(x(t) - \hat{x}(t))^T \right\} \quad (21)$$

Odhad stavu (t) je minimalizován v každém kroku dle následujícího kritéria:

$$J_{LMS}(t) = tr P(t) \quad (22)$$

V dalším kroku je předpokládáno, že v kroku t daného algoritmu je znám apriorní odhad stavu $x(t)$. Jeho podmíněná střední hodnota je značena následovně:

$$\hat{x}(t|t-1) \tag{23}$$

Kovariantní matice, která zachycuje chyby odhadu stavu, je značena následovně:

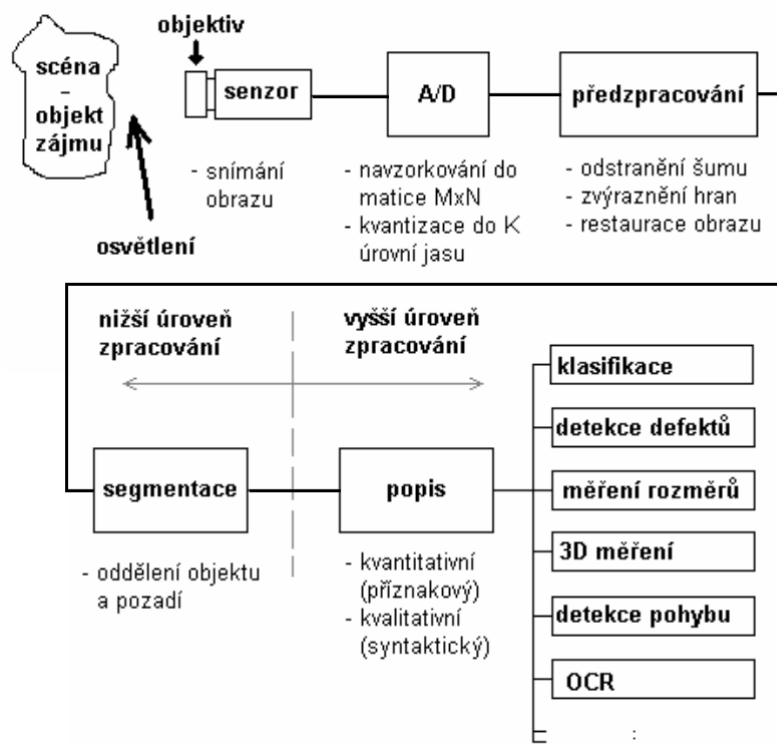
$$P(t|t-1) \tag{24}$$

Počáteční stav daného systému lze popsat pomocí kovariantní matice $\hat{P}(0) = P(0|-1)$ a střední hodnoty $\hat{x}(0) = xb(0|-1)$. Poté, co je získána hodnota výstupu $y(t)$, je nutné hodnoty aktualizovat a tím získat požadovanou kovariantní matici chyby odhadu $P(t|t)$ a aposteriorní odhad stavu $\hat{x}(t|t)$.

Dalším krokem je nalezení apriorní hodnoty daných veličin $\hat{x}(t+1|t)$ a $P(t+1|t)$ v čase $t+1$ pomocí kovariantní matice chyby odhadu a aposteriorního odhadu stavu. Tento krok lze nazvat úlohou vývoje stavu lineárního stochastického systému (Havlena, Štecha, 1999).

2.4 Zpracování obrazu

Při procesu zpracování obrazu dochází ke zpracování obrazových informací. Tento proces lze rozdělit do série dílčích úkonů, viz Obrázek č. 10.



Obrázek 10: Řetězec procesů, který slouží pro zpracování digitálního obrazu. Upraveno ze zdroje: (Horák, Kalová, Petyovský, Richter, 2008)

Předzpracování obrazu

Předzpracování obrazu je proces, jehož cílem je potlačit nedostatky, které mohly nastat během procesu digitalizace nebo během přenášení obrazu. Jedná se zejména o potlačení šumu v obraze, odstranění zkreslení obrazu nebo potlačení či zvýraznění rysů obrazu. Vstupem a výstupem tohoto procesu je vždy obraz (Horák, Kalová, Petyovský, Richter, 2008):

$$g(i, j) \quad \text{Element vstupního obrazu} \quad (25)$$

$$f(i, j) \quad \text{Element transformovaného obrazu} \quad (26)$$

Obraz může obsahovat následující vady:

Šum

Může být důsledkem digitalizace nebo může nastat během přenosu obrazu. Šum se dále dělí do následujících skupin:

- **Bílý šum** - Je využíván pro simulování nejhorších degradací v obraze.
- **Gaussův šum** - Aproximace degradace obrazu.
- **Aditivní šum** - Šum, který vzniká během snímání či během přenosu obrazu.
- **Kvantizační šum** - Nastává pokud není použit dostatečný počet úrovní jasu během kvantizace (Horák, Kalová, Petyovský, Richter, 2008).

Zkreslení

Zkreslení obrazu je jev, který může nastat zejména z důvodu:

- Souběžným pohybem snímače a snímaného objektu
- Nevhodným zaostřením
- Vadou optické soustavy
- Nelinearitou opticko-elektrického senzoru

Pro předzpracování požadovaného obrazu, jehož cílem je odstranění výše zmíněných nedostatků, můžeme použít metody rozdělené do těchto skupin:

Jasové transformace

Tento typ transformací je založen na principu změny hodnot u obrazové funkce obrazu dle stanoveného pravidla. Výsledkem této transformace je obraz, který má identické rozlišení a bitovou hloubku se vstupním obrazem. Tato transformace se provádí při nerovnoměrném osvětlení obrazu nebo také pro korekci kontrastu. Jasové transformace lze rozdělit do skupin podle velikosti okolí analyzovaného bodu (Horák, 2010):

- **Globální transformace** - Novou hodnotu bodu vypočítáme ze všech bodů ve vstupním obraze. Do této skupiny patří např. metoda integrálního obrazu.
- **Lokální transformace** - Nová hodnota bodu je určena na základě okolí analyzovaného bodu např. lokální filtrování.
- **Bodové transformace** - Hodnota bodu ve výstupním obraze vychází pouze z jednoho bodu. Jedná se např. o jasovou korekci.

Geometrické transformace

Metoda, která transformuje souřadnice bodů na základě mapování bodů ze vstupního obrazu na body výstupního obrazu. Pokud body nenáleží do ortogonálního rastru, tak je provedena pro tyto body interpolace jasových hodnot. Tato transformace může být použita, pokud došlo ke zkreslení obrazu během jeho pořizování nebo pokud byla provedena geometrická změna na obraze.

Segmentace obrazu

Segmentace je proces, při kterém jsou extrahovány objekty od pozadí. Tento proces patří mezi jeden z nejdůležitějších úkonů v rámci analýzy obsahu dat v obraze. Cílem segmentace je rozdělení vstupního obrazu do celků, které mají značnou souvislost s předměty reálného světa v obraze. Výsledkem tohoto procesu je množina oblastí, které se navzájem nepřekrývají a které odpovídají jednoznačně či částečně objektům v daném obraze. Podle tohoto můžeme dále dělit segmentaci na částečnou a kompletní segmentaci (Hájovský, Pustková, Kutálek, 2012).

Zde jsou uvedeny vybrané segmentační techniky, které byly použity v rámci práce:

Prahování

Prahování je metoda, která patří mezi nejstarší a nejjednodušší metody pro segmentaci. Metoda je široce využívána i na vzdory svému omezení. Mezi hlavní výhody metody patří její jednoduchost a nízká časová složitost. Hlavní myšlenka metody vychází z toho, že objekty a pozadí obrazu mají rozdílnou intenzitu jasu. Na základě stanovení úrovně intenzity též. práh dokáže metoda rozhodnout pro každý pixel, zda se jedná o pixel objektu k segmentaci či pozadí (Hájovský, Pustková, Kutálek, 2012).

$$g(x, y) = \begin{cases} 1 & \text{prof}(x, y) > T \\ 0 & \text{prof}(x, y) \leq T \end{cases} \quad (27)$$

Pixely, které mají hodnotu 0, patří do pozadí. Pixely s hodnotou nula patří k objektům. Prahování dále můžeme dělit do těchto skupin:

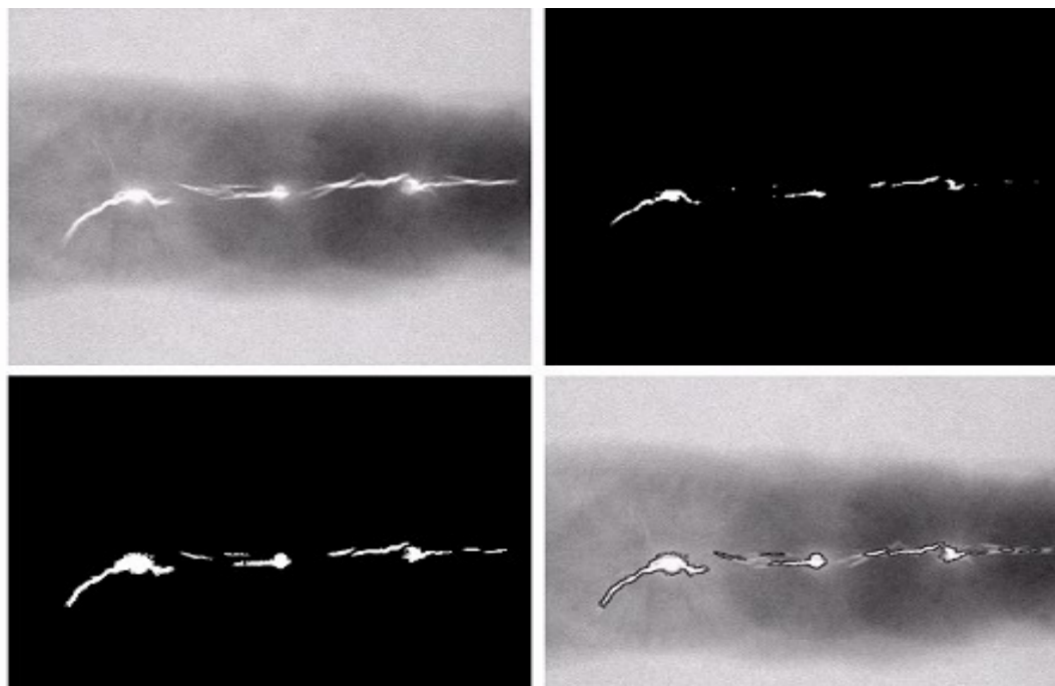
- **Globální prahování** - Nastavá, pokud práh T závisí jen na $f(x, y)$.
- **Lokální prahování** - Práh závisí na $f(x, y)$ a $p(x, y)$.
- **Adaptivní prahování** - Práh T závisí na prostorových souřadnicích (Hájovský, Pustková, Kutálek, 2012).

Narůstání oblastí

Segmentační metoda, která pracuje s malými oblastmi nebo jednotlivými pixely. Tyto oblasti jsou pomocí algoritmu dále spojovány do větších struktur. Tento postup je aplikován, dokud v obraze existují oblasti, které lze sloučit dohromady. Tato metoda využívá v počátku vybranou množinu bodů, obvykle zvolenou na základě přirozenosti úlohy. Na základě této množiny bodů a předdefinovaných kritérií dochází k rozrůstání oblastí. Kritéria, podle kterých jsou dané pixely přiřazovány do oblastí, jsou vybrány na základě vlastností obrazu (Hájovský, Pustková, Kutálek, 2012).

Vybrané metody popisu obrazu

V této kapitole jsou uvedeny vybrané deskriptory pro popis obrazu. Příznaky, které byly získány použitím deskriptorů, jsou použity jako vstupní data pro natrénování vhodných klasifikátorů umožňujících detekovat hledané objekty v obraze.



Obrázek 11: Ukázka segmentačního procesu. Zdroj: (Hájovský, Pustková, Kutálek, 2012)

Histogram orientovaných gradientů

Metoda, která dokáže popsat objekty pomocí intenzity a směru gradientů bez známé polohy objektů. Vstupem mohou být barevné obrazy i obrazy v odstínech šedi. Metoda tedy popisuje objekty pomocí distribuce gradientů složených z velikostí a svého směru (Dallal, Triggs, 2005).

Algoritmus je složen z následujících kroků:

1. Stanovení gradientů

Tento krok stanoví gradienty dle Gaussova filtru s maskou $[-1,0,1]$ a $[-1,0,1]$ transponovaně. Výstupem jsou dva obrazy pro souřadnice x a y .

2. Výpočet velikosti gradientu

Pro každý pixel je stanovena vzdálenost gradientu tj. vzdálenost x,y a směr gradientu.

3. Konstrukce histogramu orientací gradientů

Stanoví se velikost výseče nejčastěji 180 (pro lidi) nebo 360 stupňů (pro vozidla) a počet s , který rozděluje danou výseč na stejně velké části.

4. Normalizace histogramu

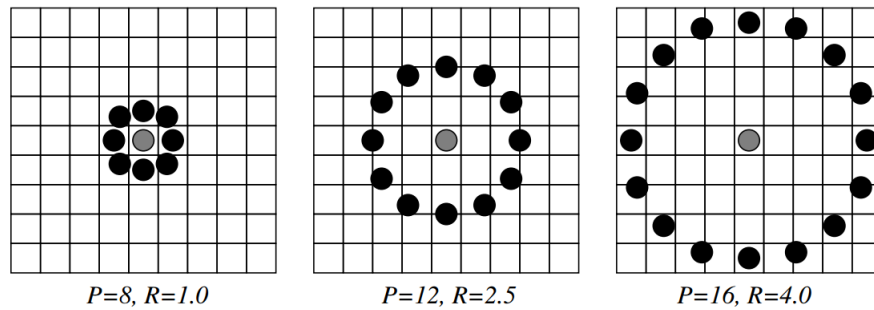
5. Konstrukce deskriptoru

Nejprve se rozdělí rovnoměrně vstupní obraz do čtvercových bloků. Bloky jsou dále rozděleny do buněk, pro které jsou vypočteny normované histogramy ori-

entací. Histogramy buněk jsou zprůměrovány a tvoří popis HOG bloku neboli HOG deskriptoru. Blok se poté přesune a proces je opakován znovu. Výsledek je zřetězení všech bloků v obraze, což je sestavený HOG detektor (Dallal, Triggs, 2005).

Lokální binární vzory

Metoda, která je určena k popisu vlastností obrazu dle příznaků. Během vytváření příznaku je iterativně procházen každý pixel v obraze, kromě krajních pixelů. Pro každý pixel je pomocí funkce ohodnocení stanovena jeho hodnota, viz Rovnice č. 28 a č. 29. Funkce ohodnocení je ovlivněna parametrem R , který určuje požadovanou vzdálenost pixelů od daného pixelu a parametrem P určující počet pixelů v okolí pixelu (MÄENPÄÄ, 2003).



Obrázek 12: Ukázka lokálních binárních vzorů. Zdroj: (MÄENPÄÄ, 2003)

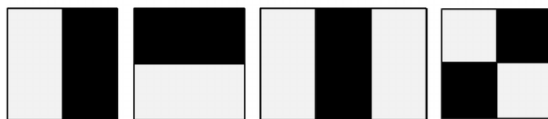
$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^p \quad (28)$$

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (29)$$

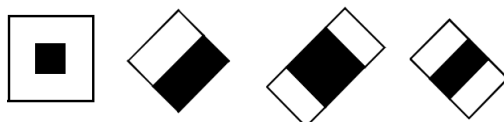
Haar-like příznaky

Haarovy příznaky jsou jednoduché příznaky založeny na principu Haarovy vlnky, které jsou tvořeny dvěma či více definovanými oblastmi. Příznaky využívají rozdílů jasů mezi oblastmi. Odezva příznaku je dána rovnicí č. 30, kde hodnoty R_{White} a R_{Black} reprezentují sumy intenzity pixelů v bílé či černé oblasti.

$$F_{\text{Haar}} = E(R_{\text{White}}) - E(R_{\text{Black}}). \quad (30)$$



Obrázek 13: Základní příznaky: a, b) hranové příznaky jsou tvořeny dvěma oblastmi. c) čárový příznak tvořen třemi oblastmi. d) diagonální příznak je tvořen čtyřmi oblastmi. Upraveno ze zdroje: (Králík, 2012)



Obrázek 14: Rozšířené příznaky. Upraveno ze zdroje: (Králík, 2012)

Základní příznaky, které jsou zobrazeny v obrázku č. 13, lze dále rozdělit do skupin na hranové, středové a čárové příznaky. Ze základních příznaků lze odvodit další typy příznaků, které obecně zvyšují přesnost detekce, jejichž výběr je znázorněn v obrázku č. 14 (Králík, 2012).

Optický tok

Podstata optického toku je popsána v sekci 2.1.

Vybrané metody klasifikace v obraze

V této kapitole jsou uvedeny vybrané metody, které jsou určeny pro klasifikaci dat získaných z předchozích kroků zpracování obrazu. Tyto metody jsou používány pro klasifikaci objektů v obraze.

- **AdaBoost**

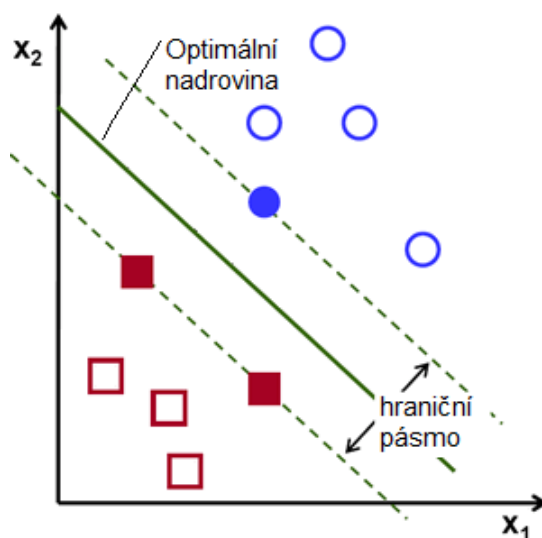
Těž adaptivní boosting je algoritmus strojového učení, který je používán pro klasifikaci a regresní analýzu. Algoritmus byl navržen v roce 1995 Y. Freundem a R. Schapirem. Obecně platí, že metoda je více odolná k přeučení než většina algoritmů strojového učení, ale je více náchylná na zašumělé vstupní data. Metoda je nazývána adaptivní, jelikož využívá vícenásobné iterace k vytvoření jediného silného klasifikátoru pomocí iterativního přidání slabých klasifikátorů. Během každé iterace učícího procesu je přidán nový slabý klasifikátor a zároveň je upraven váhový vektor, který je změněn na základě špatně klasifikovaných vzorů v předchozí iteraci. Výsledkem je klasifikátor, který disponuje vyšší přesností než slabé klasifikátory (Schapire, 2013).

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t \cdot h_t(x)\right) \quad (31)$$

Kde $H(x)$ je finální silný klasifikátor. $h_t(x)$ reprezentují slabé klasifikátory (Schapire, 2013).

- **Metoda podpůrných vektorů**

Metoda patří do skupiny metod strojového učení s učitelem. Metodu lze použít pro klasifikaci dat do vybraných tříd či pro regresní analýzu. Podstatou úlohy je hledání optimální nadroviny, která separuje prostor příznaků takovým způsobem, že třídy vstupních dat leží ve vlastních poloprostorech, viz Obrázek č. 15.



Obrázek 15: Rozdělení vstupních dat dle optimální nadroviny. Upraveno ze zdroje: (OpenCV 2.4, 2014)

Optimální rovina vytváří hraniční pásmo, které vytváří okolo nadroviny nejširší pásmo bez bodů. Pro popis nadroviny jsou použity body, které leží na kraji pásma, též nazývány podpůrné vektory (angl. Support vectors) (OpenCV 2.4, 2014).

- **Umělá neuronová síť**

Neuronová síť je algoritmus, který je založen na principu fungování lidského mozku, je tedy inspirován biologickou neuronovou sítí. V biologické neuronové síti jsou základním prvkem nervové buňky, též nazývány neurony. Tyto neurony jsou navzájem spojeny hranami, které jsou ohodnoceny váhami. Tato spojení a schopnost adaptovat váhy dle trénovacích vzorů ve vstupních datech poskytuje nové možnosti v analýze dat (StatSoft, 2013).

Neuronové sítě lze využít při řešení následujících úloh:

- **Regrese**

Regresní analýza je zaměřena na predikci spojitých proměnných na základě vstupních dat.

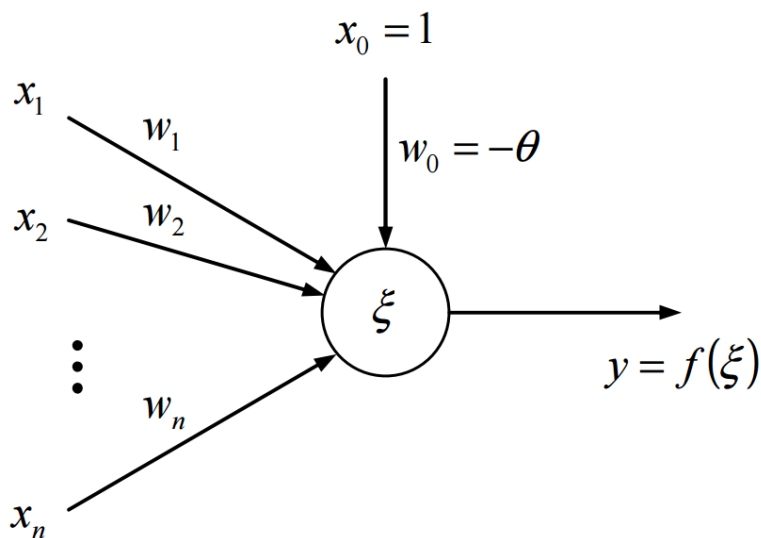
– **Klasifikace**

Pomocí úrovně cílové proměnné a kombinace vstupních dat je vytvořen model, který je schopen klasifikovat nově získaná data do tříd.

– **Shluková analýza**

Je analýza, která nevyužívá závislou proměnnou, patří tedy do algoritmů učení bez učitele. Cílem analýzy je nalezení shluků ve vstupních datech. Jedná se například o Kohenovu síť

Model umělého neuronu je tvořen vstupní, výstupní a funkční částí. Dle stanovených vah jsou jednotlivé vstupní data zvýhodněny či potlačeny. Funkční část neuronu je zaměřena na zpracování informací ze vstupních dat a poté generuje výstup. Výstupní část přenáší získanou informaci na vstup dalších neuronů v síti. Pokud je suma vstupů neuronu χ_i , které jsou vynásobeny konkrétními vahami w_i vyšší než hodnota prahu θ , tak je spočítán výstup neuronu $f(\xi)$. Neuron může být formulován následovně $y = f\left(\sum_{i=1}^n x_i \cdot w_i - \theta\right)$. Přičemž platí, že χ_i je hodnota na i -tém vstupu a w_i určuje váhu i -tého vstupu. Celkový počet vstupů udává proměnná n . f je transformační funkce, která počítá hodnotu výstupu y , viz Obrázek č. 16, (StatSoft, 2013).

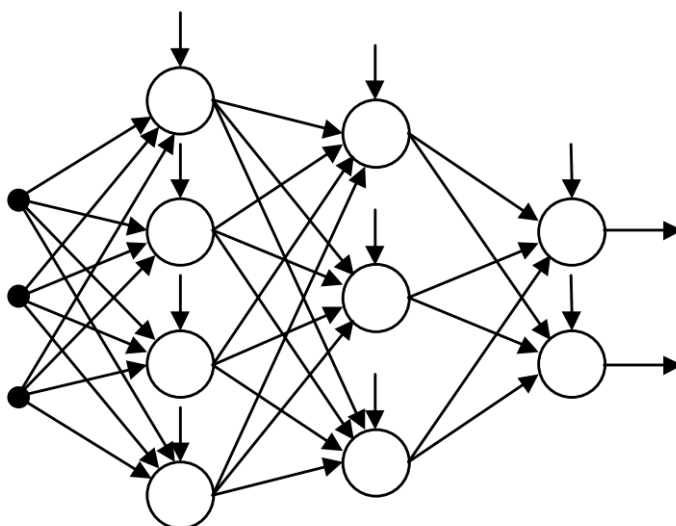


Obrázek 16: Schéma umělého neuronu. Zdroj: (StatSoft, 2013)

Jeden umělý neuron není schopen provádět o moc sofistikovanější funkce než regresní analýza. Podstata umělé neuronové sítě je tedy propojení více neuronů do komplexnějších struktur. Jednotlivé neurony jsou strukturovány do vrstev, viz Obrázek č. 17.

Díky tomuto složitému spojení dokáže síť nalézt složitější nelineární vztahy ve vstupních datech. Nevýhodou neuronové sítě je neschopnost interpretace, proč u

daného vstupu došlo k takovému výsledku. Nelze tedy získat elementární předpis závislosti mezi závislou proměnnou a nezávislými proměnnými (StatSoft, 2013).



Obrázek 17: Topologie třívrstvé umělé neuronové sítě. Zdroj: (StatSoft, 2013)

Pokud je neuronová síť složena z malého počtu neuronů tak obecně platí, že je její schopnost popsat závislosti na vstupních datech je slabší. Pokud síť obsahuje příliš mnoho neuronů, tak je pravděpodobné, že síť nebude mít problém s reprezentací závislosti ve vstupních datech. Ale je více pravděpodobné, že síť bude mít horší schopnost generalizace, tedy horší odhad správného výsledku na nově pořízených datech. Tento jev je nazýván přeučení neuronové sítě. Na přeučení sítě může mít vliv příliš velký počet vstupních parametrů nebo menší počet vstupních dat modelu. Cílem je tedy určitý kompromis mezi trénovacím výkonem a schopností generalizovat znalosti na nově získaných datech. Tento problém je spojen s vhodným rozdělením vstupních dat (StatSoft, 2013).

Data mohou být rozdělena na trénovací, testovací a případně validační množinu:

- **Trénovací množina**
Náhodně zvolená část vstupních dat, která je použita pro natrénování sítě.
- **Testovací množina**
Další zvolená část dat, která slouží k zastavení trénovacího procesu, aby nenastalo přetrénování sítě.
- **Validační množina**
Zbývá část vstupních dat, která má za cíl ověřit finální kvalitu modelu.

Zbývá část vstupních dat, která má za cíl ověřit finální kvalitu modelu. Typicky se toto rozdělení dělá v poměru 50-25-25, případně 70-15-15. Ve výsledcích je

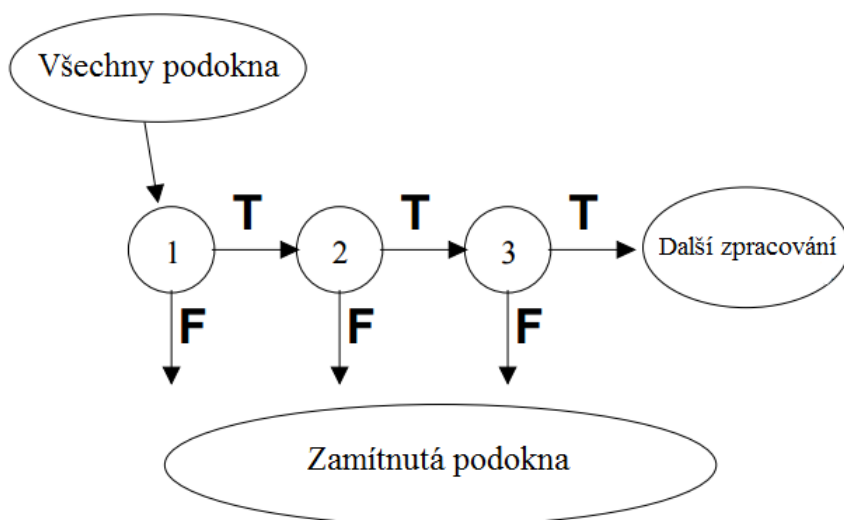
pak reportován výkon na každé z těchto množin, přičemž většinou vybíráme model, který nemá příliš velké výkyvy mezi výkony na jednotlivých množinách (StatSoft, 2013).

2.5 Známé detektory osob

Viola/Jones detektor

P.Viola a M.Jones navrhli v roce 2001 detektor objektů, který otestovali pomocí detekce obličejů v obraze. Detektor využívá reprezentaci obrazu pomocí metody integrální obrazu, která umožňuje detektoru vyhodnocovat požadované příznaky mnohem rychleji. Pro další zrychlení detektoru je použit princip kaskádového klasifikátoru.

Detektor pracuje s množinou Haarových příznaků s prahovou hodnotou, které slouží jako jednoduché klasifikátory. Detektor dále využívá klasifikační metodu strojového učení AdaBoost, též adaptivní boosting pro složení jednoduchých klasifikátorů do kaskádového klasifikátoru podle pozitivních a negativních snímků. Metoda AdaBoost vybírá malý počet zásadních příznaků z větší množiny dat (Viola, Jones, 2001).



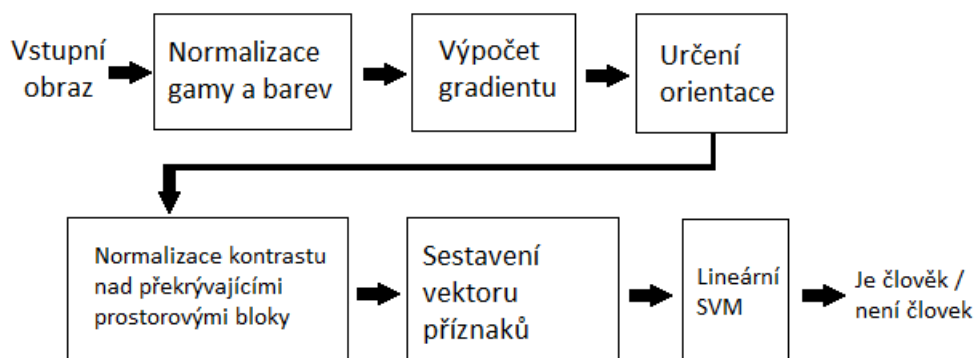
Obrázek 18: Kaskádový klasifikátor. Upraveno ze zdroje: (Viola, Jones, 2001)

Celý proces detekce lze tedy popsat rozhodovacím stromem též kaskádou, která je složena z jednoduchých klasifikátorů a má za úkol eliminovat negativní podokna, viz Obrázek č. 18. Výsledkem kaskády je množina pozitivních oken.

Dalal/Triggs detektor

Detektor byl vytvořen v roce 2005 a slouží pro detekci osob v obraze. Detektor využívá histogramy orientovaných gradientů, jejichž princip byl vysvětlen výše. Získané deskriptory, které byly vytvořeny z HOG příznaků, jsou využívány lineární

metodou podpůrných vektorů. Tento jednoduchý klasifikační model, který spadá do oblasti strojového učení s učitelem, je natrénován a optimalizován pomocí získaných vstupních dat. Detektor je poté schopen s určitou pravděpodobností detekovat osoby v obraze. Celý postup detektoru je znázorněn na obrázku č. 19.



Obrázek 19: Řetězec ukazující princip detektoru. Upraveno ze zdroje: (Dallal, Triggs, 2005).

2.6 Charakteristiky vyhodnocení

Klasifikační problém, který používá dvě třídy, lze popsat formálně následovně. Každá instance I je namapována k jednomu prvku množiny p, n , která obsahuje pozitivní a negativní označení třídy. Klasifikační model mapuje instance do predikovaných tříd. Některé klasifikační modely mají spojitý výstup. Na základě tohoto výstupu mohou být aplikovány různé hraniční hodnoty oddělující tyto třídy. Ostatní klasifikační modely produkují diskrétní výstupy, které označují třídy (Fawcett, 2004).

| | | Skutečná třída | |
|----------|----------|-----------------|-----------------|
| | | p | n |
| Predikce | Y | True Positives | False Positives |
| | N | False Negatives | True Negatives |
| Součet: | | P | N |

Obrázek 20: Matice záměn (angl. Confusion matrix), která zachycuje skutečnou a predikovanou třídu. Upraveno ze zdroje (Fawcett, 2004)

Pokud je instance pozitivní a je také klasifikována jako pozitivní, tak je instance započtena do skupiny pravda - pozitivní (angl. true positive). Pokud je instance pozitivní a je klasifikována jako negativní, tak je označena jako nepravda - negativní (angl. false negative). Jestli je instance negativní a je klasifikována jako negativní, tak patří do skupiny pravda - negativní (angl. true negative). Pokud by v tomto případě byla instance klasifikována pozitivně tak patří do skupiny nepravda - pozitivní (angl. false positive) (Fawcett, 2004).

Dle této matice lze stanovit metriky pro hodnocení klasifikačních modelů. Zde jsou uvedeny vybrané metriky:

- **Pokrytí (angl. Recall nebo True positive rate)**

Metrika, která vyjadřuje s jakou pravděpodobností bude nalezen objekt v obraze. Je definována následovně:

$$\text{Pokrytí} = \frac{TP}{P} \quad (32)$$

- **Přesnost (angl. Precision)**

Metrika uvádí pravděpodobnost, která určuje, že oblast, která byla detekována jako objekt, je doopravdy objektem. Je definována podle následující rovnice:

$$\text{Přesnost} = \frac{TP}{FP + TP} \quad (33)$$

- **F-score**

Metrika vyjadřuje jakou přesnost má měření. Je definována následovně:

$$\text{F-score} = 2 \cdot \frac{\text{Přesnost} \cdot \text{Pokrytí}}{\text{Přesnost} + \text{Pokrytí}} \quad (34)$$

- **Frekvence výskytu FP (angl. False positive rate)**

$$\text{False positive rate} = \frac{FP}{N} \quad (35)$$

Tyto metriky byly převzaty ze zdroje (Fawcett, 2004).

- **Miss rate**

Metrika uvádí s jakou pravděpodobností nebude objekt nalezen v obraze. Jedná se tedy o protiklad k metrice Recall. Metriku lze vypočítat podle rovnice:

$$\text{Miss rate} = \frac{FN}{TP + FN} \quad (36)$$

- **FPPI**

Podíl počtu nepravda-pozitivní a celkového počtu pohybujících oken. Lze vyjádřit následovně:

$$\text{FPPI} = \frac{FP}{\text{Počet oken}} \quad (37)$$

Pomocí výše zmíněných metrik lze vytvořit hodnotící křivky. Detekování objektů je vícekrát opakováno s rozdílnými parametry vybraného detektoru např. práh. Z těchto hodnot lze sestavit tyto křivky:

- **FPPI křivka**

Křivka zachycuje závislost metriky Miss rate na FPPI. Ideální tvar křivky se blíží bodu $[0, 0]$.

- **Precision – Recall křivka**

Křivka zachycuje závislost metriky přesnosti na pokrytí. Ideální tvar křivky se blíží bodu $[1, 1]$.

- **F-score křivka**

Křivka zachycuje závislost F-score a metriky pokrytí. Ideální tvar křivky se blíží bodu $[1, 1]$. Převzato ze zdroje (Klos, 2014).

2.7 Vybrané nástroje

Knihovna Emgu CV

EmguCV je otevřená multiplatformní knihovna, která je určena zejména pro zpracování obrazu a počítačové vidění. Knihovna EmguCV poskytuje vybranou funkcionalitu knihovny OpenCV pro framework .NET (Shi, 2013). Velkou výhodou této knihovny je uspokojivý výkon, který byl ověřen pomocí testů porovnávající vybrané knihovny, viz Tabulka č. 1.

Tabulka 1: Výkonnost jednotlivých knihoven. Zdroj: (Shi, 2013)

| Kód | Knihovna | Převod do stupňů šedi [ms] | Binarizace [ms] |
|-----|-------------------|----------------------------|-----------------|
| C | OpenCV | 9.3432 | 6.9332 |
| C# | AForge.NET | 32.6548 | 19.8743 |
| C# | Emgu CV | 11.2369 | 9.6853 |
| C# | OpenCV (P/Invoke) | 10.2355 | 8.0332 |
| C# | Custom method | 33.6742 | 17.2009 |

Framework Accord.NET

Framework Accord.NET je používán zejména pro vědecké výpočty na platformě .NET. Framework obsahuje několik knihoven, které jsou zaměřeny na aplikaci vědeckých výpočtů např. zpracování statistických dat, strojové učení, rozpoznání vzorů, počítačové vidění a zpracování obrazu. Framework poskytuje velké množství rozdělení pravděpodobností, hypotézní testy, jádrové funkce a podporu pro známé měřicí techniky (Accord.NET, 2015).

3 Vstupní data analyzovaného problému

Tyto data jsou rozděleny do dvou množin podle účelu svého použití. První množina dat obsahuje videosoubory, které byly pořízeny kamerami od výrobce Basler. Druhá množina dat je vytvořena z pořízených videosouborů a slouží pro vybrané metody strojového učení a také pro testování detektorů osob.

3.1 Popis pořízených videí

Videosoubory byly pořízeny kamerou Basler ac1600-uv165 ze snímkovou frekvencí 20 fps a kamerou Basler acA2000-165uc s frekvencí snímků 10 fps. Videosoubory zachycují simulovaný a reálný pohyb osob v laboratoři Q11, která sloužila pro simulaci kancelářských prostorů. Další informace o pořízených videích jsou uvedeny v tabulce 2. Tato množina videosouborů slouží zejména pro následující účely:

1. Vytvoření datových sad
2. Testování algoritmů pro detekci dynamických objektů
3. Testování detekce a analýzy trajektorií pohybu osob
4. Zobrazení vhodných výstupů výsledného algoritmu

Tabulka 2: Vlastnosti videosouborů, které byly pořízeny kamerami od výrobce Basler.

| ID | Délka videa | Počet snímků | Rozlišení | FPS | Kamera |
|----|-------------|--------------|-----------|-----|----------------------|
| 1 | 01:13:18 | 43980 | 2040x1086 | 10 | Basler acA2000-165uc |
| 2 | 00:43:48 | 52560 | 1624x1234 | 20 | Basler acA1600-20uc |
| 3 | 01:09:05 | 41725 | 2040x1086 | 10 | Basler acA2000-165uc |

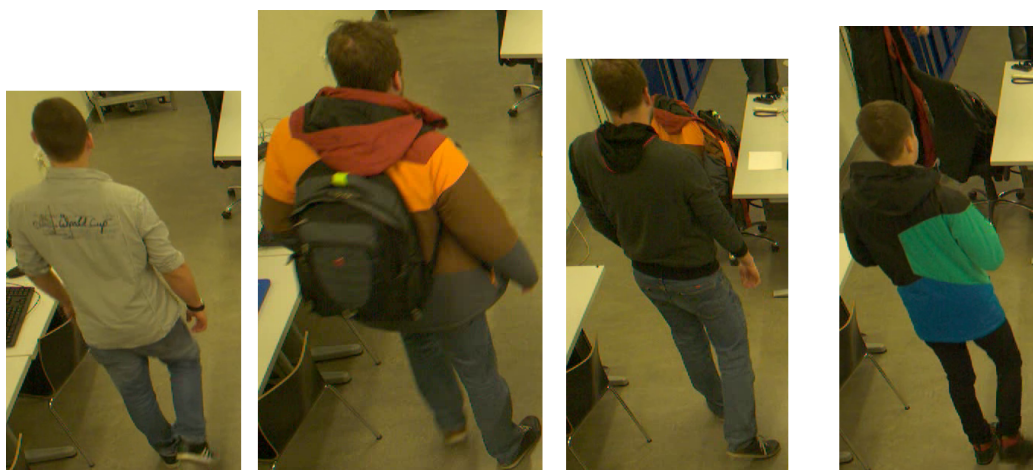
3.2 Popis datových sad

Každá datová sada obsahuje množinu snímků, které jsou ohodnoceny pozitivně či negativně. Ohodnocení každého snímku je stanoveno podle obsahu daného snímku, pokud snímek obsahuje celé tělo člověka, tak je snímek ohodnocen pozitivně tedy hodnotou 1. Pokud snímek neobsahuje celé tělo člověka nebo zachycuje jiné dynamické objekty či více osob velmi blízko u sebe, tak je snímek ohodnocen negativně pomocí hodnoty 0. V rámci práce byla přidána do vytvořené aplikace funkcionalita, která umožňuje vytvářet a upravovat datové sady na základě vstupního videosouboru. Popis jednotlivých datových sad a jejich parametrů je uveden v tabulce č. 4.

Celý proces sběru a zpracování dat lze rozdělit do následujících kroků:

1. Extrakce snímků ze vstupního videosouboru

Extrakce probíhá na základě procházení celého videosouboru, kdy je náhodně zvolen skok mezi jednotlivými snímky. Toto opatření zajistí, aby výsledná množina snímků neobsahovala redundantní nebo velmi navzájem podobné snímky. Algoritmus vyhledává v každém snímku objekty s využitím algoritmu optického toku $TV-L^1$. Ukázka extrahovaných snímků je zobrazena na obrázku č. 21.



Obrázek 21: Výběr snímků, které byly pořízeny algoritmem bez úpravy.

Tabulka 3: Porovnání rychlosti extrakce snímků z videosouboru pomocí sériového a paralelního zpracování na chipu Nvidia Geforce GTX 1060.

| | Celkový čas z 50 měření [ms] | Průměrný čas z 50 měření [ms] |
|------------------|-------------------------------------|--------------------------------------|
| Sériově | 224385 | 4487 |
| 5 vláken | 117558 | 2351 |
| 10 vláken | 102659 | 2053 |
| 25 vláken | 106679 | 2133 |
| | Průměrné vytížení CPU [%] | Průměrné využití paměti [MB] |
| Sériově | 72.2% | 550 MB |
| 5 vláken | 84.7% | 1036 MB |
| 10 vláken | 91.7% | 1328 MB |
| 25 vláken | 90.5% | 1292 MB |

Takto získaný obraz je zpracován pomocí metody prahování s více prahy a metody rozrůstání oblastí. Ze zpracovaného snímku jsou extrahovány objekty, které jsou poté uloženy ve formě snímků na disk. Algoritmus byl implementován pomocí jak sériového, tak také paralelního přístupu s použitím více vláken. V

rámci práce byl proveden test, který stanovil optimální hodnoty algoritmu, viz Tabulka č. 3. Nejlepších výsledků dosáhl algoritmus s použitím 10 vláken a použitím CUDA technologie TV- L^1 . Algoritmus dosáhl 2,18* větší rychlosti než sériový algoritmus, ale došlo k navýšení průměrného vytížení CPU a paměti.

2. Úprava snímků do jednotného formátu

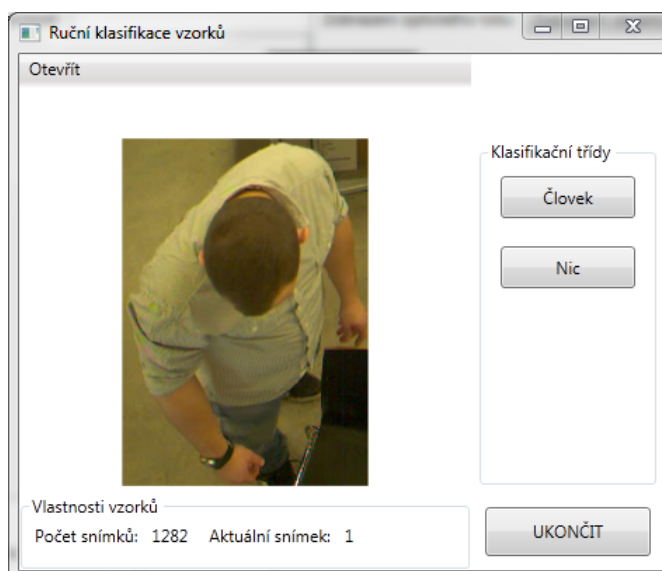
Úprava každého snímku je důležitým krokem, protože snímky musí mít jednotnou velikost, aby je bylo možné použít pro natrénování, optimalizaci a testování vybraných metod strojového učení. V rámci tohoto kroku bylo vytvořeno několik množin snímků s různou velikostí, což v rámci práce sloužilo k optimalizaci strojových metod. Dále je nutné převést snímky z RGB modelu do barev v odstínech šedi. Ukázka výsledku tohoto kroku je zobrazena na obrázku č. 22.



Obrázek 22: Výběr pozitivně ohodnocených vzorků po úpravě ve velikosti 80x60px.

3. Ruční klasifikace dat

Výsledné množiny upravených snímků musí být ručně ohodnoceny člověkem. Každý snímek je uživatelem ohodnocen na základě svého obsahu pozitivně či negativně. Na obrázku č. 23 je zobrazeno grafické rozhraní aplikace, která umožňuje klasifikovat množinu snímků.



Obrázek 23: Ukázka přidané funkcionality pro ruční klasifikaci dat.

4. Normalizace dat

Normalizace je proces, při kterém jsou data převedena pomocí normalizačních funkcí na jednotný interval hodnot. Data pro tuto práci nebyla normalizována z důvodu toho, že jednotlivé pixely snímků mohou nabývat hodnot na intervalu 0 – 255. A také proto, že se v datové sadě nevyskytly žádné extrémní hodnoty.

5. Rozdělení dat

Ve finálním kroku už je provedeno samotné rozdělení datové sady do třech částí na trénovací, cross-validační a testovací množiny. Datové sady byly rozděleny v poměru 70-15-15 podle obecného doporučení, jak bylo řečeno v teoretické části.

Tabulka 4: Popis vlastností datových sad, které jsou využívány metodami strojového učení a detektory osob.

| Název | Velikost snímků[px] | Počet snímků | Negativní/pozitivní |
|--------------|---------------------|--------------|---------------------|
| Dataset č. 1 | 80x60 | 2288 | 851/1437 |
| | 55x40 | 2288 | 851/1437 |
| | 30x20 | 2288 | 851/1437 |
| Dataset č. 2 | 80x60 | 2652 | 1058/1594 |
| | 55x40 | 2652 | 1058/1594 |
| | 30x20 | 2652 | 1058/1594 |
| Dataset č. 3 | 80x60 | 3216 | 1219/1997 |
| | 55x40 | 3216 | 1219/1997 |
| | 30x20 | 3216 | 1219/1997 |
| Dataset č. 4 | 80x60 | 3248 | 1144/2104 |
| | 55x40 | 3248 | 1144/2104 |
| | 30x20 | 3248 | 1144/2104 |



Obrázek 24: Vybrané pozitivní snímky z datové sady v rozlišení 55x40 pixelů zobrazené v odstínech šedi.

Výsledkem tvorby datových sad jsou čtyři sady, které obsahují snímky ve třech různých velikostech. Ukázka vybraných pozitivních snímků v rozlišení 55x40 pixelů je zobrazena v obrázku č. 24. Tyto sady byly dále použity pro natrénování, optimalizaci a testování vybraných klasifikačních modelů. Datové sady také slouží pro testování vybraných detektorů osob.

4 Komparace vybraných metod

4.1 Komparace metod pro výpočet optického toku

Vybrané metody pro výpočet optického toku a odečet pozadí byly porovnávány podle dvou kritérií, časové náročnosti algoritmu a také dle přesnosti detekce dynamických objektů v obraze. V rámci komparace byly testovány a porovnávány metody optického toku $TV-L^1$, Lucas-Kanade, Farnebackova a Broxova metoda a metody MOG a GMG pro odečet pozadí. Testování metod dle časové náročnosti bylo měřeno v milisekundách, které byly potřebné pro výpočet dynamického pohybu v obraze. Metody byly testovány na videosouborech s rozdílným rozlišením z důvodu, aby bylo možné porovnat, jak rozlišení může ovlivnit přesnost detekce a časovou náročnost algoritmů.



Obrázek 25: Zobrazení pohybu mezi snímky.

Pro každou metodu a rozlišení bylo provedeno 1000 testů, které počítaly počet milisekund, které byly potřeba pro výpočet optického toku mezi dvěma po sobě jdoucími snímky zachycující pohybuující se objekty, viz Obrázek č. 25. Výsledný počet milisekund, který udává časovou náročnost algoritmu, byl stanoven podle aritmetického průměru těchto testů, viz Tabulka č. 5.

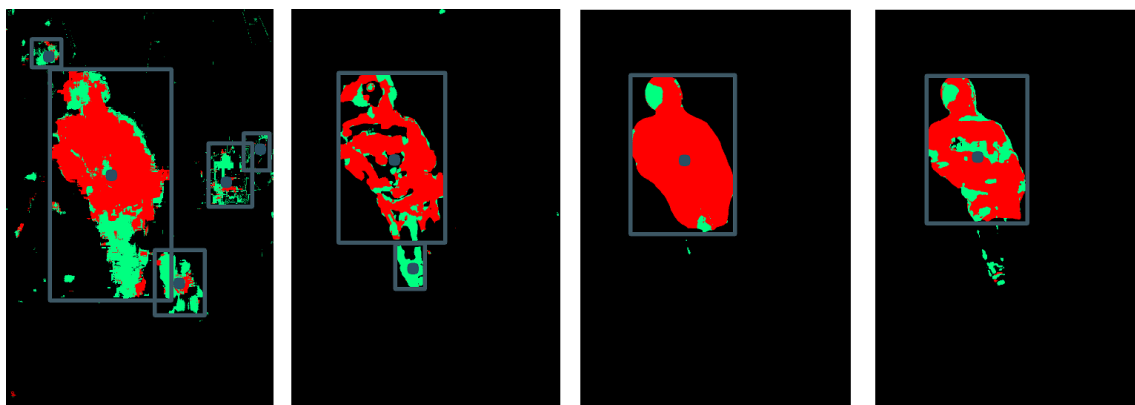
Tabulka 5: Porovnání časové náročnosti algoritmů s použitím CUDA technologie na chipu NVidia GeForce GTX 1060.

| Rozlišení | $TV-L^1$ | LK | Farneback | Brox | MOG | GMG |
|-----------|----------|--------|-----------|---------|---------|---------|
| 1920×1080 | 11244 ms | 982 ms | 1972 ms | 1128 ms | 1061 ms | 2005 ms |
| 1280×720 | 7543 ms | 542 ms | 1130 ms | 789 ms | 659 ms | 1256 ms |
| 640×480 | 2672 ms | 282 ms | 641 ms | 378 ms | 312 ms | 658 ms |
| 480×360 | 1320 ms | 189 ms | 542 ms | 292 ms | 198 ms | 497 ms |

Stanovení přesnosti detekce bylo měřeno dle měřítka Miss rate, který udává procentuální počet pohybujících se objektů, které nebyly detekovány nebo pokud jejich výsledná detekce byla velmi rozdílná od reálné podoby objektu, viz Obrázek č. 26. V rámci testování přesnosti detekce bylo také provedeno 1000 testů pro každou metodu, rozlišení a z jejich výsledků byla stanovena průměrná procentuální hodnota měřítka Miss rate pro každou metodu a rozlišení, viz Tabulka č. 6.

Tabulka 6: Porovnání přesnosti detekce pomocí Miss rate měřítka s použitím CUDA technologie na chipu NVidia GeForce GTX 1060.

| Rozlišení | TV- L^1 | LK | Farneback | Brox | MOG | GMG |
|-----------|-----------|-----|-----------|------|-----|-----|
| 1920×1080 | 3% | 21% | 8% | 5% | 5% | 2% |
| 1280×720 | 6% | 22% | 12% | 10% | 5% | 3% |
| 640x480 | 7% | 24% | 13% | 14% | 7% | 4% |
| 480x360 | 9% | 27% | 15% | 17% | 8% | 6% |



Obrázek 26: Porovnání detekce pohybu pomocí algoritmu Lucas-Kanade, Farneback, Brox, TV- L^1 .

Výsledky testování ukazují, že změna rozlišení má velký vliv na změnu časové náročnosti metod, ale také je patrné, že změna rozlišení nemá tak výrazný vliv na správnou detekci objektů. Při volbě vhodné metody je důležité se rozhodnout, zda bude kladen důraz na vysokou rychlost metody nebo na přesnost detekce dynamických objektů. Metody pro odečet pozadí poskytují vysokou přesnost a nízkou časovou náročnost, nicméně výsledkem těchto metod je pouze digitální obraz, který neposkytuje informace o samotném pohybu jako je tomu u metod optického toku, které poskytují vektorové pole. V rámci této práce je použita pro detekci dynamických objektů metoda TV- L^1 .

4.2 Komparace metod pro klasifikaci obrazu

Vybrané klasifikační metody a známé detektory osob v obraze byly porovnány na testovacích sadách dle přesnosti. Tyto metody jsou využívány v rámci procesu verifikace k ověření, zda trajektorie obsahuje osobu či ne. Testované algoritmy lze tedy rozdělit do následujících skupin:

Známé detektory osob

První skupina metod obsahuje známé detektory osob v obraze, které dokáží na základě vstupního obrazu detekovat pozici, výšku a šířku osob v obraze. V rámci této skupiny byl testován detektor Viola/Jones a HOG detektor. Testování probíhalo na datové sadě č. 4 v rozlišení 80x60 pixelů. Vybrané detektory postupně detekovali všechny snímky z datové sady a poté byla stanovena jejich přesnost detekce, viz Tabulka č. 7.

Tabulka 7: Výsledky, které byly dosaženy vybranými detektory během testovacího procesu.

| Detektory | TP | FN | FP | TN | Pokrytí | Přesnost |
|----------------------|-----|-----|-----|----|---------|----------|
| Detektor Viola/Jones | 49 | 211 | 171 | 56 | 0,19 | 0,22 |
| HOG+SVM detektor | 163 | 97 | 137 | 90 | 0,64 | 0,54 |

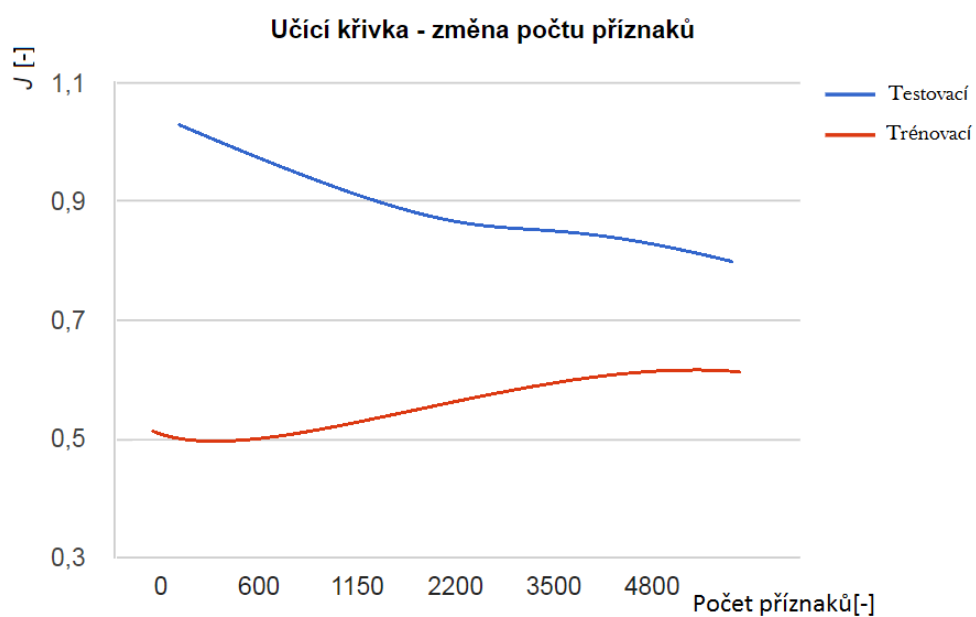
Z výsledků testu je patrné, že detektor Viola/Jones není schopen detekovat osoby v trajektoriích ve většině případů a je tedy téměř nepoužitelný pro verifikaci trajektorií. HOG detektor prokázal v testu relativně dobrou schopnost detekce osob a má tedy výraznou převahu nad detektorem Viola/Jones. HOG detektor je možné použít pro verifikaci nalezených trajektorií.

Vybrané klasifikační modely

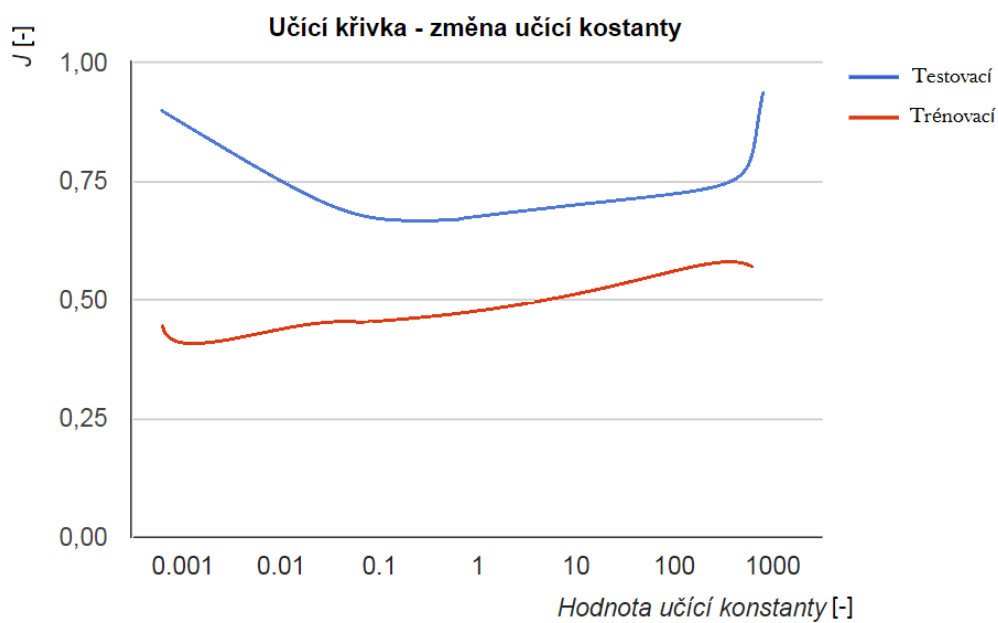
Do druhé testované skupiny metod patří vybrané klasifikační binární modely, které mohou s určitou pravděpodobností klasifikovat obraz do dvou tříd podle toho, zda snímek obsahuje člověka či ne. Do této testované skupiny patří vícevrstvá umělá neuronová síť, adaptivní boosting a lineární SVM. Trénování, optimalizace a testování klasifikačních modelů bylo provedeno na základě pořízené datové sady č. 4 v rozlišení 80x60 pixelů, která obsahuje nejrepresentativnější vzorky dat. Výsledky testu jsou zobrazeny v tabulce č. 8.

Vícevrstvá umělá neuronová síť

Neuronová síť byla natrénována pomocí datové sady č. 4 ve čtyřiceti iteracích pomocí učicího algoritmu Levenberg-Marquardt. Topologie neuronové sítě byla vytvořena pomocí tří vrstev a dvaceti skrytých neuronů. Byly také vykresleny učicí křivky, které sloužily pro optimalizaci neuronové sítě a pomohly stanovit optimální hodnotu počtu příznaků, učicí konstantu a počet vzorků.

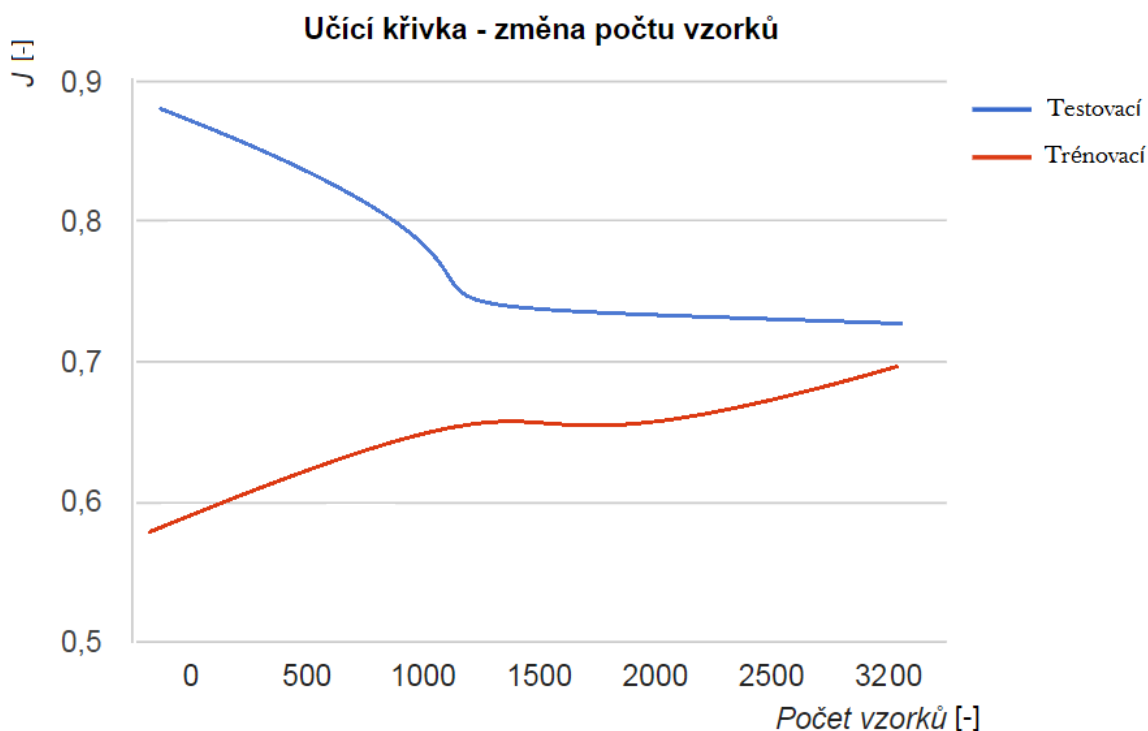


Obrázek 27: Učící křivky, které ukazují změnu v závislosti na počtu příznaků datové sady.



Obrázek 28: Učící křivky, které ukazují změnu v závislosti na hodnotě učící konstanty.

Optimální počet příznaků je stanoven na hodnotu 4800. Pokud by byl vybrán menší počet, tak by mohlo nastat přeučení resp. vysoká variance. Pokud by byla zvolena vyšší hodnota, tak by mohlo dojít k podučení sítě, viz Obrázek č. 28. Optimální počet vzorků je stanoven na hodnotu 2500. Při této hodnotě nenastává přeučení ani podučení, jak vyplynulo z grafu, viz Obrázek č. 29.



Obrázek 29: Učící křivky, které ukazují změnu v závislosti na počtu vzorků datové sady.

Optimální hodnota učící konstanty je stanovena na hodnotu jedna. Z grafu vyplývá, že při této hodnotě nenastává přeučení ani podučení sítě, viz Obrázek č. 28.

Lineární SVM

je využívána lineární metoda, tak není použita žádná jádrová funkce. Dále byl nastaven parametr, který určuje počet tříd do kterých bude metoda klasifikovat vzorky na hodnotu dva.

Adaptivní boosting

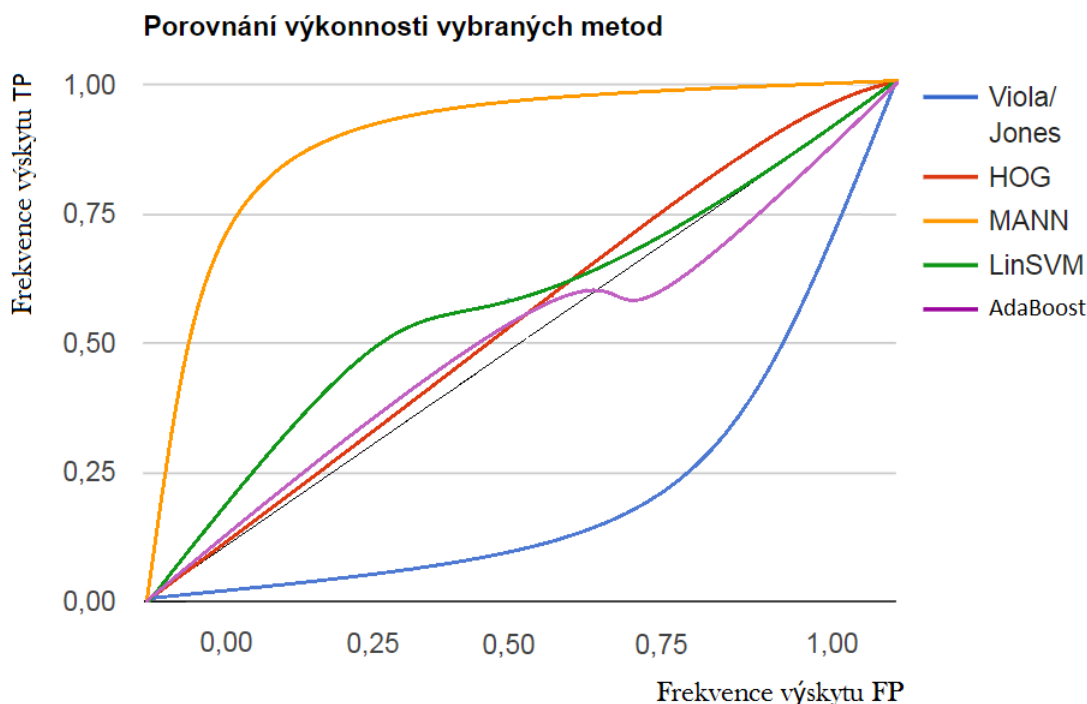
Algoritmus AdaBoost byl natrénován pomocí datové sady č. 4. Dále byly nastaveny vstupní parametry algoritmu. Počet slabých klasifikátorů byl stanoven na hodnotu 100. Hodnota prahu pro snížení výsledného času výpočtu je 0.95. Dalším parametrem je typ boostovacího algoritmu, který byl stanoven na Real AdaBoost. Real AdaBoost je vhodný pro klasifikaci dat do tříd.

Nejlepšího výsledku ve skupině klasifikačních modelů dosáhla vícevrstvá neuronová síť, která prokázala vysokou přesnost při klasifikaci a její použití pro verifikaci

Tabulka 8: Dosažené výsledky vybraných klasifikačních modelů, které byly získány během testovacího procesu.

| Klasifikační modely | TP | FN | TN | FP | Pokrytí | Přesnost |
|--------------------------|-----|-----|-----|-----|---------|----------|
| Lineární SVM | 154 | 106 | 117 | 110 | 0,61 | 0,57 |
| Vícevrstvá neuronová síť | 238 | 22 | 23 | 104 | 0,91 | 0,92 |
| Adaptivní boosting | 157 | 103 | 134 | 93 | 0,6 | 0,0,53 |

je tedy vhodné. Metody lineární SVM a AdaBoost dosáhly také relativně dobrých výsledků a je tedy možné je použít pro proces verifikace.



Obrázek 30: ROC křivky zobrazující výkonnost jednotlivých metod, které byly naučeny, optimalizovány a testovány na datové sadě č. 4 s rozlišením snímků 80x60.

Vícevrstvá umělá neuronová síť dosáhla v testu nejlepší schopnosti klasifikovat objekty trajektorií do správných tříd a je tedy nejlepší volbou pro verifikaci trajektorií. Proto byla použita jako výchozí metoda verifikace ve vytvořené aplikaci, nicméně aplikace umožňuje použít všechny výše popsané metody pro verifikaci. Z testování je patrné, že HOG detektor, AdaBoost a lineární SVM vykazovaly také dobré výsledky klasifikace a jejich použití pro verifikaci je vhodné. Viola/Jones detektor dosáhl v testu horších výsledků a proto jeho použití pro verifikaci není vhodné. Jednotlivé ROC křivky klasifikátorů a detektorů osob jsou zobrazeny v obrázku č. 30.

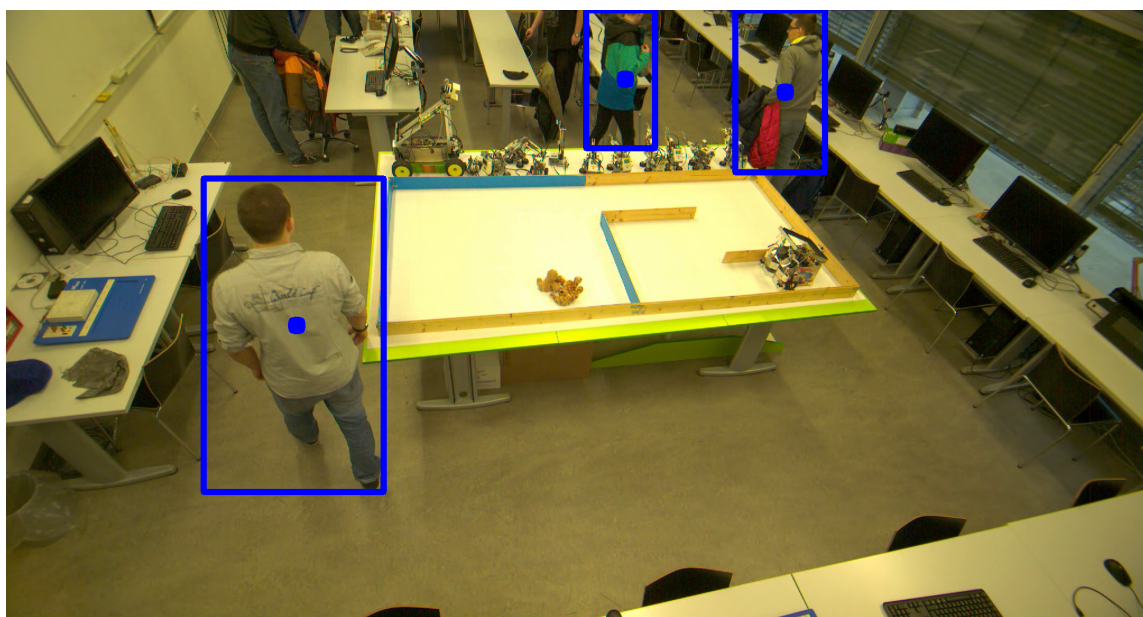
5 Návrh metodiky vyhodnocení pohybu osob

5.1 Metodika detekce trajektorií

Hlavním účelem algoritmu je nalezení dynamických objektů napříč jednotlivými snímky videosouboru či množiny snímků. Tyto nalezené objekty jsou poté spárovány do vhodných trajektorií pohybu, které reprezentují pohyb dynamických objektů na celém intervalu videosouboru nebo na jeho vybrané části. Celý proces tvorby trajektorií pohybu osob lze rozdělit do následujících kroků:

Detekce dynamických objektů v obraze

Podstatou tohoto procesu je nalezení dynamických objektů v obraze pomocí algoritmu TV- L^1 . Tyto nalezené objekty jsou poté extrahovány z obrazu na základě algoritmu rozrůstání oblastí a metody prahování. V dalším kroku jsou tyto objekty filtrovány dle stanovených vlastností, aby došlo k eliminaci objektů, které nemají dostatečnou velikost a hustotu. Ukázka detekce dynamických objektů je zobrazena na obrázku č. 31.

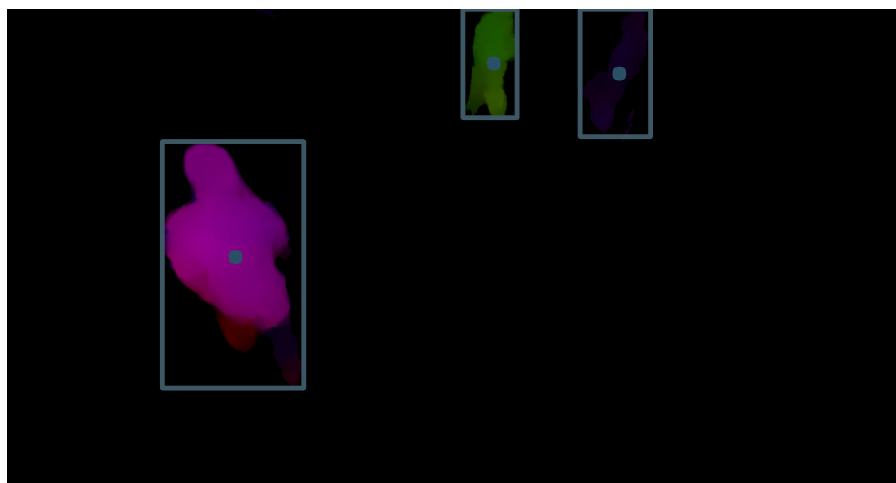


Obrázek 31: Výchozí snímek zobrazující nalezené dynamické objekty

Celý tento proces lze rozdělit do následujících kroků:

1. Detekce dynamických objektů

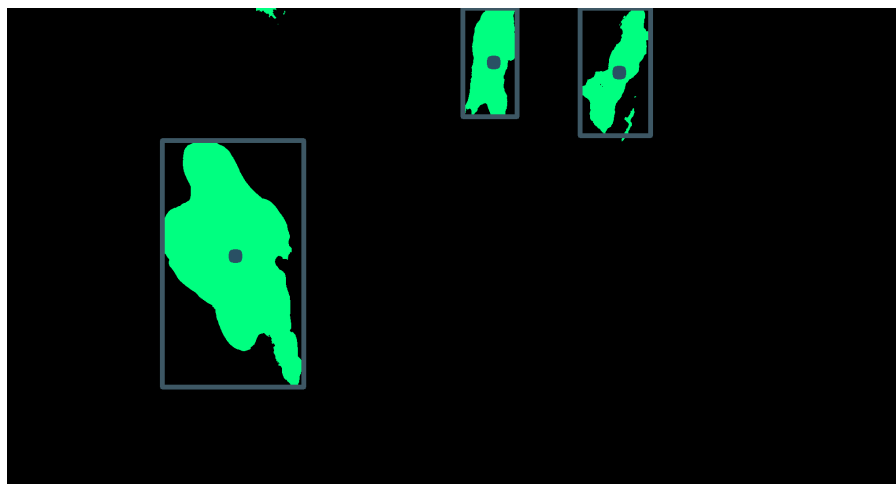
Tento krok vypočítá pomocí algoritmu optického toku TV- L^1 pohyb dynamických objektů, který je reprezentován maticemi směrových vektorů. Tyto vektory jsou dále převedeny do barevného modelu HSV podle své délky, která udává rychlost pohybu a podle svého směru, viz Obrázek č. 32.



Obrázek 32: Obraz, který obsahuje optický tok zobrazený pomocí modelu HSV.

2. Extrakce dynamických objektů z obrazu

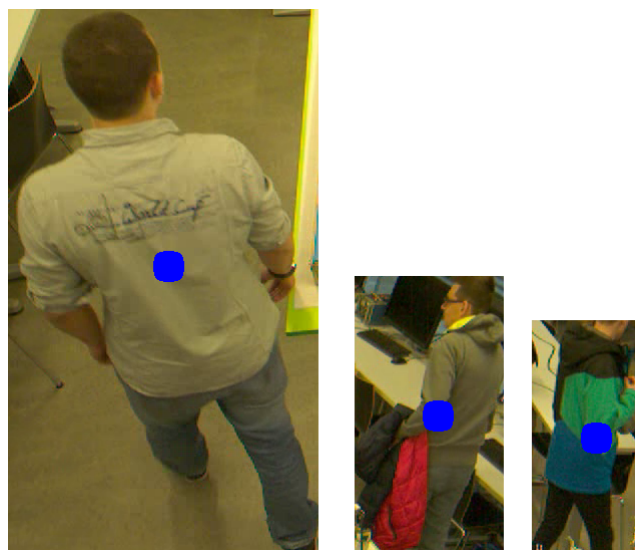
V tomto kroku se obraz, který obsahuje optický tok zobrazený v HSV modelu, převede pomocí metody prahování s více prahy do obrazu s homogenními barvami, viz Obrázek č. 33. Na takto zpracovaný obraz už lze aplikovat metodu rozrůstání oblastí, která extrahuje nalezené objekty. Takto získané objekty jsou poté reprezentovány pomocí binárních velkých objektů.



Obrázek 33: Obraz obsahující optický tok po aplikaci metody prahování s více prahy.

3. Filtrace nalezených objektů

Důležitým krokem je filtrace extrahovaných objektů dle hustoty a velikosti. Objekty, které nemají dostatečnou hustotu, tedy obsahují malý optický tok, jsou vyloučeny. Objekty, které mají velmi rozdílnou šířku oproti výšce nebo jsou velmi malé, jsou vyloučeny z dalšího zpracování, viz Obrázek č. 34.



Obrázek 34: Nalezené objekty, které jsou filtrovány dle hustoty a své velikosti.

Spárování objektů do trajektorií pohybu

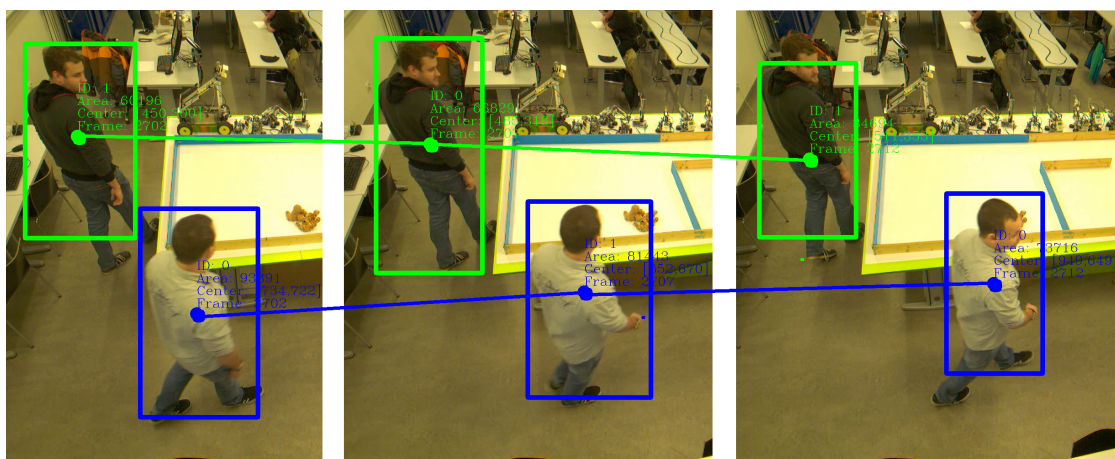
Tento proces spáruje množinu dynamických objektů, které byly detekovány v minulém kroku, napříč jednotlivými snímky. Dynamické objekty jsou v tomto kroku spárovány na základě maďarské metody, která spáruje objekty vždy z dvou po sobě jdoucích snímků, viz Obrázek č. 36. Maďarská metoda používá těžiště dynamických objektů ke stanovení optimálních párů. Výsledkem tohoto procesu je množina trajektorií pohybu dynamických objektů.



Obrázek 35: Zobrazení rádiusu každého objektu pomocí kružnic.

U každého nalezeného objektu je stanoven rádius, který udává pomocnou oblast

pro spárování, viz Obrázek č. 35. Pokud jsou detekovány objekty v náhodném snímku a následující snímek obsahuje objekty, jejichž těžiště spadají do pomocné oblasti, tak jsou objekty spárovány. Pokud těžiště objektů nespádají do daných oblastí, tak jsou vytvořeny nové trajektorie a pro stávající trajektorie je provedena predikce na základě Kalmanova filtru.



Obrázek 36: Spárování objektů napříč třemi snímky pomocí maďarské metody.



Obrázek 37: Spárování objektů napříč třemi snímky pomocí maďarské metody s využitím kalmanova filtru pro predikci polohy objektu.

Kalmanův filtr je dále používán v situacích, kdy není možné detekovat objekt a je nezbytné mít alespoň přibližnou polohu objektu. Tato situace může nastat například, pokud jsou dva či více objektů velmi blízko sobě a jsou proto detekovány jako jeden celek, viz Obrázek č. 37 nebo pokud objekt zastaví a nevykonává žádný pohyb.

5.2 Metodika analýzy trajektorií

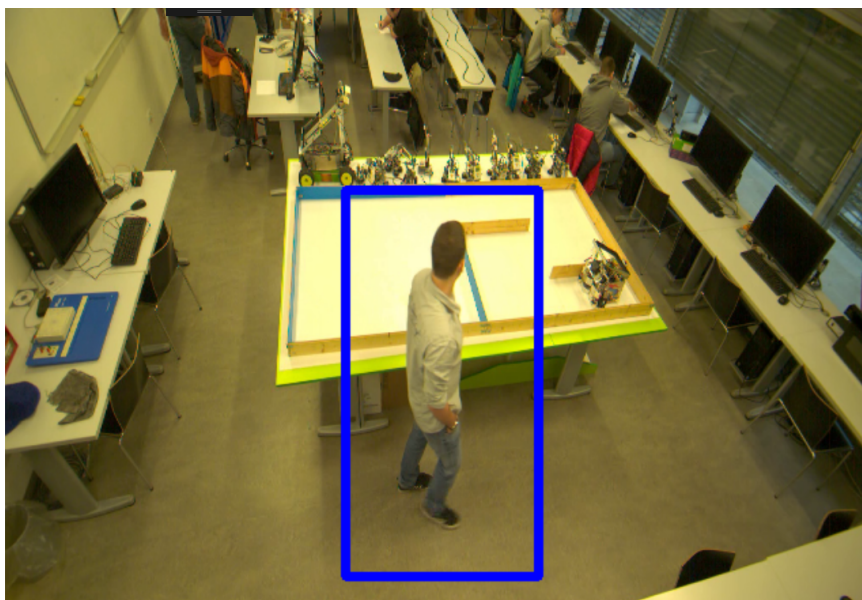
Cílem algoritmu je provedení analýzy trajektorií pohybu, které byly získány algoritmem pro detekci trajektorií z videosouboru či množiny snímků. Analýza je složena ze dvou hlavních kroků tj. verifikační proces a analýza vlastností nalezených trajektorií.

Verifikace trajektorií

Verifikace osob je proces, který je proveden až po nalezení všech trajektorií pohybu dynamických objektů ve vybraném časovém úseku videosouboru nebo na jeho celé délce. Tento proces má za úkol ověřit, zda dané trajektorie pohybu byly opravdu vytvořeny osobami, což má za důsledek eliminaci těch trajektorií, které patří jiným dynamickým objektům s podobnou velikostí jako člověk. V rámci verifikace byly ověřeny dvě skupiny metod, které mohou v určité míře odhadnout, zda je daný objekt člověk či ne.

Princip dle detektorů osob

Princip verifikace pomocí vybraného detektoru je založen na tom, že detektor provede detekci osob v deseti náhodně vybraných snímcích trajektorie. Pozice detekovaných osob ve vybraných snímcích (viz. Obrázek č. 38) musí mít přibližnou polohu s pozicemi objektů v dané trajektorii alespoň ve třech případech. Pokud je tato podmínka splněna, tak je trajektorie vyhodnocena jako pozitivní. Pokud pozice objektů v dané trajektorii neodpovídá pozicím osob nalezených pomocí detektoru ve více než sedmi případech, tak je trajektorie vyřazena z dalšího zpracování.



Obrázek 38: Detekovaný člověk v obraze pomocí detektoru Dallala/Triggse.

Princip dle vybraných klasifikačních modelů

Proces verifikace provede klasifikaci deseti náhodně vybraných objektů z každé tra-

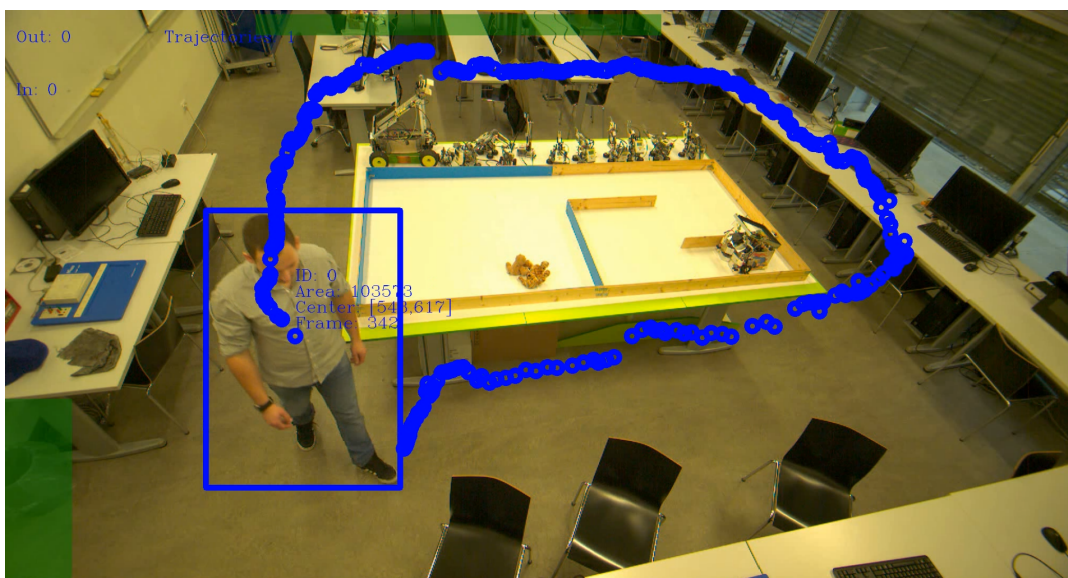
jektorie pohybu, viz Obrázek č. 39. Pokud byla klasifikace alespoň třikrát pozitivní, tak daná trajektorie patří člověku. Pokud tato podmínka není splněna, tak je trajektorie vyřazena z dalšího zpracování. Výchozí klasifikační metodou je vícevrstvá neuronová síť.



Obrázek 39: Deset vybraných objektů z trajektorie, které jsou poté klasifikovány vybraným binárním klasifikátorem.

Analýza tvaru trajektorií

Analýza tvaru je dalším krokem, který již slouží pro analyzování trajektorií pohybu osob, jelikož nežádoucí trajektorie jiných objektů byly pravděpodobně eliminovány v procesu verifikace. Analýza ověřuje tvar a směr každé trajektorie pohybu viz Obrázek č. 40. Tento proces slouží zejména pro agregaci dat z nalezených trajektorií. Na základě této analýzy se stanoví, jestli je trajektorie aktivní tj. jestli osoba je stále v obraze. Pomocí tohoto kroku je možné odhadnout aktuální počet osob v obraze. Dále je analýza schopna určit u objektů, které se již nenachází v obraze, čas odchodu či příchodu do oblasti nebo je také schopna určit rychlost pohybu osob.



Obrázek 40: Zobrazení pomocných oblastí a vypočtené trajektorie pohybu osob.

Pokud není přístup do dané oblasti (místnosti, pracovní oblasti apod.) možný z celého obvodu snímku, tak se musí před samotným zpracováním videosouboru stanovit vstupně-výstupní oblasti. Tyto vybrané oblasti slouží pro odchod či příchod

do celé oblasti, jedná se např. o vstupní dveře, vstupní prostory a místa, kde kamera nezachycuje celou plochu oblasti, ale pouze jeho část.

Výstupem analýzy je tedy množina aktivních trajektorií, která obsahuje osoby, které se pohybují v dané oblasti a u kterých není možné stanovit čas odchodu či příchodu, ale je možné stanovit určité ukazatele. Druhou množinou je množina obsahující neaktivní trajektorie osob. U těchto trajektorií je možné stanovit čas příchodu a odchodu do oblasti, a další vhodné ukazatele.

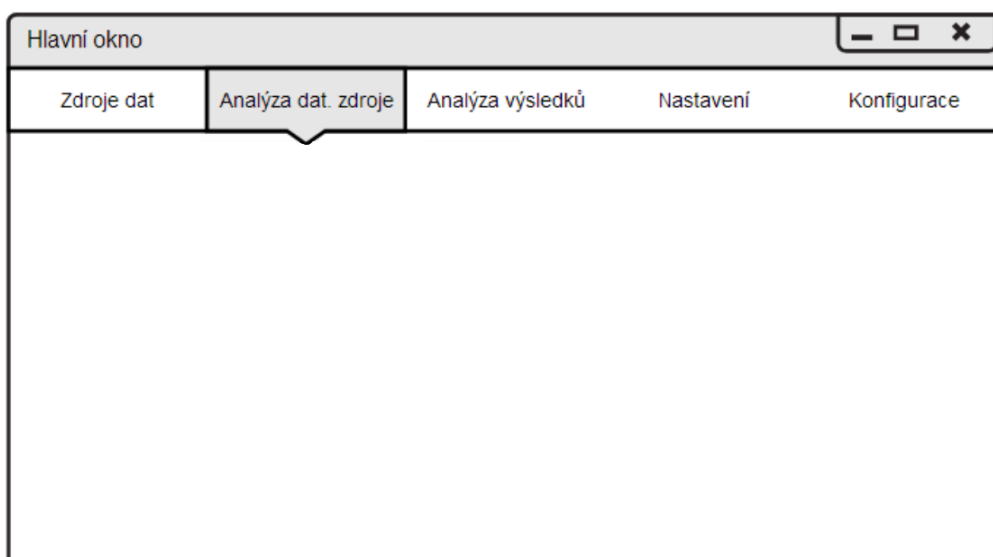
Posledním krokem analýzy je odhad počtu osob v trajektoriích. Pokud jsou po nějakou dobu objekty velmi blízko u sebe (typicky skupina osob, která komunikuje navzájem mezi sebou), tak algoritmus je schopen odhadnout počet osob v trajektorii na základě aritmetického průměru ploch objektů, které jsou obsaženy v trajektorii. Koeficient, který vznikne podílem aritmetického průměru ploch objektů v trajektorii a plochou jednoho objektu reprezentujícího osobu, udává přibližný počet osob v trajektorii. Pokud se nějaká osoba či více osob připojí nebo odpojí od dané skupiny osob, tak je ovlivněn výsledný průměr vyjadřující počet osob v dané trajektorii.

6 Tvorba řídicí aplikace

V této kapitole jsou popsány dva hlavní pohledy na navrženou řídicí aplikaci. V první části je popsán pohled z hlediska uživatele, který popisuje význačné grafické prvky aplikace. V uživatelském pohledu je také popsána základní obsluha aplikace, její funkcionality a také nástroje, které umí vhodně zobrazit či zpracovat získané trajektorie. Druhá část kapitoly je zaměřena na programátorský pohled. V této části je popsána a vysvětlena základní struktura aplikace.

6.1 Uživatelský pohled

V sekci je popsána základní manipulace s řídicí aplikací z hlediska uživatele. Je zde také popsána funkcionality, nastavení parametrů a grafické rozhraní význačných částí aplikace. Poté jsou vysvětleny jednotlivé položky v hlavním okně a jejich návaznosti na další části aplikace, viz Obrázek č. 41.



Obrázek 41: Náhled hlavního okna, který byl vytvořen pomocí nástroje Moqups.

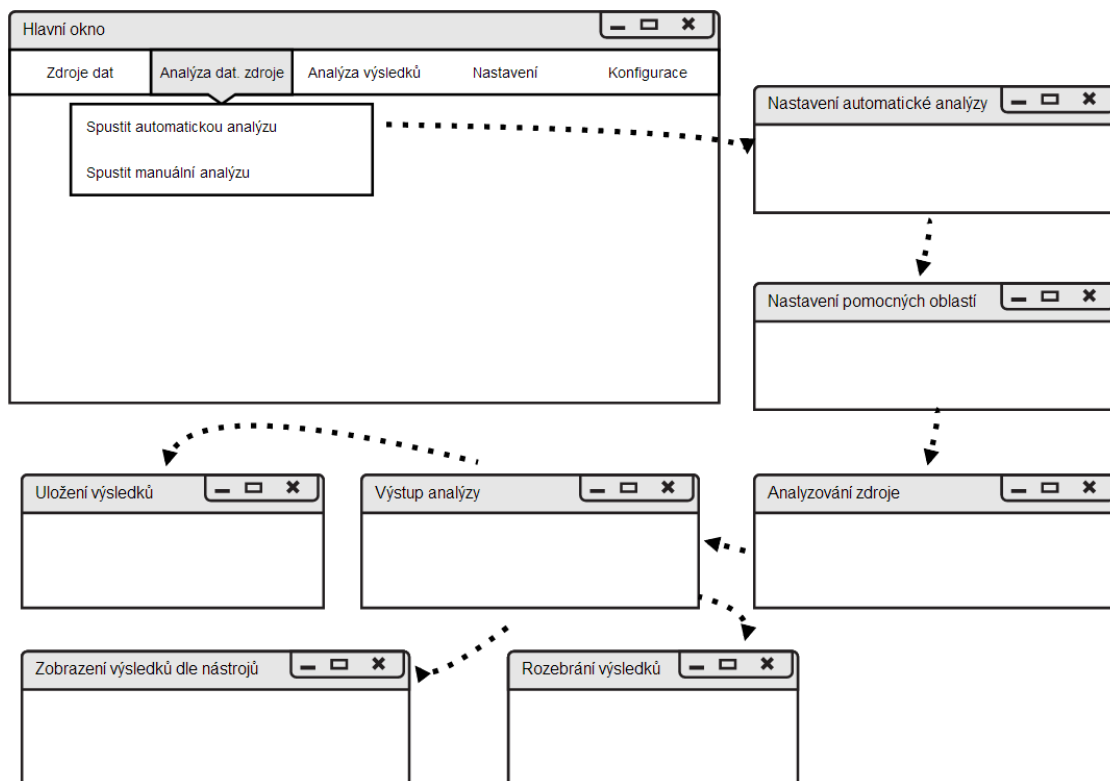
Položka - Zdroje dat

Tato položka v menu je určena pro nastavení datových zdrojů, které jsou dále zpracovávány dle navržených algoritmů. Je možné načíst různé typy videosouborů či adresáře obsahující množinu snímků.

1. Načíst videosoubor
2. Načíst množinu snímků

Položka - Analýza datového zdroje

Analýza datového zdroje je nejdůležitější částí řídicí aplikace a umožňuje spouštět navržený algoritmus, který je schopen detekovat a následně analyzovat trajektorie pohybu osob. Dále jsou popsány všechny parametry navrženého algoritmu, grafické rozhraní a návaznosti na něj. V této části jsou popsány přechody mezi jednotlivými kroky celého procesu analýzy, viz Obrázek č. 42.



Obrázek 42: Přechod vedlejších oken z hlavního okna. Náhled vytvořen pomocí nástroje Moqups.

1. Automatická analýza

Automatická analýza je stěžejní částí aplikace, protože implementuje výše popsanou metodiku pro detekci a analýzu trajektorií pohybu osob. Jednotlivé kroky, které je třeba provést pro získání relevantního výsledku, jsou rozděleny následovně:

Nastavení automatické analýzy

Účelem tohoto kroku je nastavení parametrů algoritmu, který dokáže detekovat a analyzovat trajektorie pohybu osob z vybraného či úplného časového intervalu datového zdroje. Algoritmus vyžaduje čtená nastavení parametrů, proto jsou tyto parametry a jejich doporučené hodnoty dále rozepsány v tabulce č. 1.

Tabulka 9: Popis parametrů, které mohou být nastaveny v rámci automatické analýzy.

| Parametr | Hodnoty | Doporučené hodnoty |
|----------------------|--------------------------|--------------------|
| Počáteční snímek | 0 - počet snímků videa | - |
| Konečný snímek | 0 - počet snímků vstupu | - |
| Spodní hranice prahu | 0 - 255 | 20 |
| Horní hranice prahu | 0 - 255 | 255 |
| Skok | 0 - 20 | 1 |
| Metoda detekce | Brox | - |
| | Lucas-Kanade | - |
| | Farneback | - |
| Aktivace CUDA | TV- L^1 | TV- L^1 |
| | True/False | True |
| Metoda verifikace | Viola/Jones detektor | - |
| | HOG detektor | - |
| | Vícevrstvá neuronová síť | Neuronová síť |
| | Adaptivní boosting | - |
| | Lineární SVM | - |
| | Bez verifikace | - |

Nastavení pomocných oblastí

Uživatel musí stanovit vstupně-výstupní oblasti, které zachycují oblasti umožňující vstup či odchod do snímané oblasti.

Analyzování zdroje

Tento krok provádí detekci a analýzu trajektorií pohybu osob na základě parametrů, které byly stanoveny v minulých krocích. Výsledkem je analyzovaná množina trajektorií pohybu osob.

Výstup analýzy

Trajektorie, které byly získány v předchozích krocích, mohou být zpracovány mnoha způsoby. Uživatel může v tomto kroku uložit získané informace do vhodných datových formátů nebo také může zobrazit výsledky pomocí několika způsobů.

Zobrazení výsledků dle nástrojů

Prezentační vrstva umožňuje zobrazit detekované a analyzované trajektorie pohybu pomocí různých nástrojů. Tyto nástroje dokáží podle svého účelu zobrazit získané informace v mnohem zajímavějších formách. Také je možné zvolit pouze vybrané časové intervaly videosouborů či množin snímků což umožňuje zobrazit pouze informace, které jsou důležité pro uživatele. Dále lze také filtrovat nalezené trajektorie a vybírat trajektorie, které jsou z nějakého důvodu důležité.

- **Teplotní mapy**

Mapy slouží k zobrazení četnosti výskytu pozice osob na vybraném časovém

úseku datového zdroje nebo na jeho celé délce. Mapy poskytují zajímavější pohled na získané trajektorie pohybu a mohou být použity jako podklad pro další rozhodování.

- **Grafy**

Dalším zajímavým nástrojem pro prezentaci získaných trajektorií jsou grafy. Grafy umožňují vykreslit průběh trajektorií v závislosti na čase a tím poskytují důležité informace o snímané oblasti. Aplikace umožňuje využívat tyto grafy:

- **Liniový**

Liniový graf zobrazuje aktuální predikovaný počet osob v obraze v závislosti na čase. Z tohoto grafu lze zjistit časový interval s největším či nejmenším počtem osob.

- **Sloupcový**

Sloupcový graf slouží pro zobrazení poměru příchodů a odchodů v pravidelných časových intervalech.

- **Statistické charakteristiky**

Nad získanými trajektoriemi lze vypočítat základní statistické charakteristiky např. medián či modus a další. Tyto charakteristiky mohou poskytovat informace o tom, v jakém časovém okamžiku bylo nejvíce osob v obraze apod. Charakteristiky mohou být počítány na celém intervalu nebo jen jeho vybrané části.

Uložení výsledků

Aplikace poskytuje možnost ukládat a zpracovávat data pomocí různých formátů, což umožňuje zpracování a analyzování dat v dlouhodobém měřítku a poskytuje přesnější pohled na získané informace. Dále je možné ukládat informace z vybraného časového intervalu, takže není potřeba ukládat nepotřebné či redundantní informace. Uživatel může také filtrovat nalezené trajektorie a ukládat jen potřebné trajektorie pohybu.

- **Nerelační databáze MongoDB**

Informace o nalezených trajektoriích je možné uložit a zpracovávat pomocí databáze MongoDB. V aplikaci se nastaví základní parametry pro připojení k databázovému serveru. Takto zpracované informace mohou být použity pro širší analýzu konkrétní oblasti zachycené kamerou.

- **Formát XML**

Získané trajektorie pohybu osob, které byly získány dle navrženého algoritmu, lze exportovat do formátu XML a tím uchovat dané informace pro další použití. Tyto soubory lze poté zpětně importovat do aplikace, což umožňuje provádět širší analýzu nad konkrétním datovým zdrojem a tím

lze získat komplexnější pohled na analyzovaný prostor. Ukázka vygenerovaného souboru je zobrazena na obrázku č. 43.

```
<?xml version="1.0" encoding="utf-8"?>
<Trajectories source="sourceName">
  <Trajectory id="0">
    <DetectedObject frame="252">
      <Rectangle>
        <Height>150<Height />
        <Width>128<Width />
        <X>0<X />
        <Y>0<Y />
      <Rectangle />
      <CenterX>75<CenterX />
      <CenterY>64<CenterY />
      <Area>8574<Area />
    <DetectedObject />
  <Trajectory />
</Trajectories />
<DetectedObject frame="252">
  <Rectangle>
    <Height>146<Height />
    <Width>114<Width />
    <X>10<X />
    <Y>12<Y />
  <Rectangle />
  <CenterX>83<CenterX />
  <CenterY>74<CenterY />
  <Area>8974<Area />
</DetectedObject />
</Trajectories />
```

Obrázek 43: Ukázka výsledného souboru ve formátu XML. Soubor ukládá informace o trajektoriích pohybu osob.

- **Formát JSON**

Aplikace také umožňuje export nalezených trajektorií do formátu JSON. Takto vytvořené soubory mohou být zpětně importovány a mohou být použity pro rozsáhlejší analýzu vytyčeného prostoru. Ukázka vygenerovaného souboru je zobrazena na obrázku č. 44.

```
{
  "0": [
    {
      "Key": 100,
      "Value": {
        "Rectangle": "10, 10, 100, 100",
        "Image": null,
        "Center": "10,10",
        "Area": 100
      }
    }
  ],
  "1": [
    {
      "Key": 200,
      "Value": {
        "Rectangle": "10, 10, 100, 100",
        "Image": null,
        "Center": "10,10",
        "Area": 100
      }
    }
  ],
  "1": [
    {
      "Key": 1200,
      "Value": {
        "Rectangle": "10, 10, 100, 100",
        "Image": null,
        "Center": "210,210",
        "Area": 1200
      }
    },
    {
      "Key": 2200,
      "Value": {
        "Rectangle": "10, 10, 100, 100",
        "Image": null,
        "Center": "210,210",
        "Area": 1200
      }
    }
  ]
}
```

Obrázek 44: Ukázka výsledného souboru ve formátu JSON. Soubor ukládá informace o trajektoriích pohybu osob.

Rozebrání výsledků

Informace, které byly získány pomocí detekce a analýzy trajektorií pohybu osob z konkrétního datového zdroje, mohou být dále detailně rozebrány. V tomto kroku je možné projít všechny snímky z datového zdroje a zobrazit k nim vypočtené vlastnosti, které byly získány navrženým algoritmem. Dále je možné filtrovat a zpětně analyzovat vybrané trajektorie pohybu nebo lze také zobrazit vypočtené vlastnosti pro různé zobrazení optického toku.

2. Manuální analýza

Tato analýza umožňuje detekci objektů pomocí výpočtu optického toku mezi dvěma snímky. Nalezené objekty lze zobrazit pomocí různých zobrazení optického toku. Použití je vhodné zejména pro testování algoritmu či optimalizaci parametrů automatické analýzy.

Položka - Analýza výsledků

Pomocí této položky v menu může uživatel importovat informace o trajektoriích osob pro konkrétní snímaný prostor. Lze importovat soubory ve formátu JSON a XML, nebo lze načíst informace z nerelační databáze MongoDB.

Položka - Nastavení

Tato položka umožňuje nastavovat parametry dílčích algoritmů v rámci celé aplikace. Lze nastavit parametry pro algoritmy optického toku nebo odečít pozadí, Kalmanův filtr a další.

Položka - Konfigurace

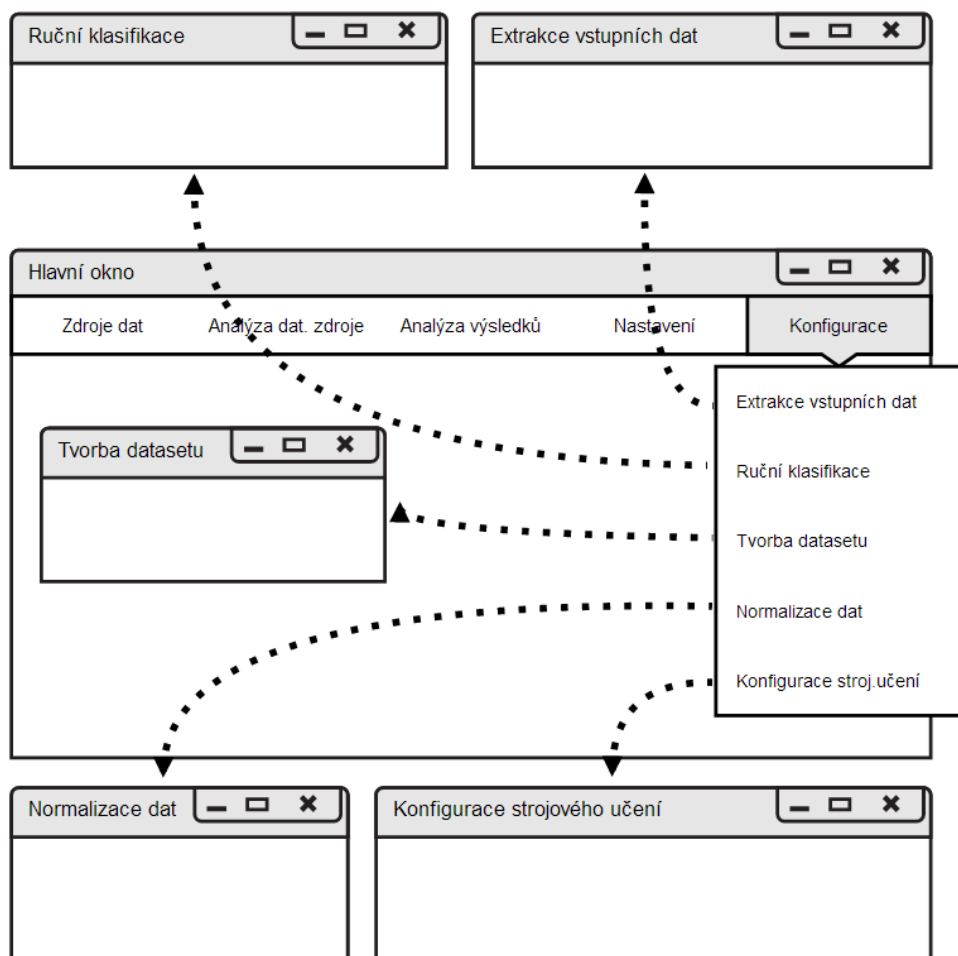
Položka konfigurace je primárně určena pro extrakci dat z vybraného zdroje dat, manipulaci a úpravu datových sad a zejména pro konfiguraci vybraných metod strojového učení. Celý proces je rozdělen do pěti samostatných grafických oken, které obsluhují konkrétní část procesu, viz Obrázek č. 45. Jedná se zejména o tyto položky:

1. Extrakce vstupních dat

Tato část aplikace umožňuje uživateli extrahovat pravděpodobné snímky dynamických objektů z vybraného zdroje dat. Uživatel zde může nastavit několik důležitých parametrů, které ovlivňují získané výsledky. Jedná se zejména o parametr skok, který určuje minimální vzdálenost mezi dvěma snímky a tím zajišťuje eliminaci velmi podobných snímků dynamických objektů. Dalšími parametry je hledaná minimální výška a šířka, které umožňují extrahovat pouze objekty, které mají velikost podobnou člověku. Extrahování snímků může být provedeno buď sériovým či paralelním zpracováním a výsledné snímky jsou uloženy do konkrétního adresáře.

2. Ruční klasifikace

Ruční klasifikace umožňuje klasifikovat snímky dynamických objektů, které byly získány v předchozím kroku. Uživatel na základě obsahu snímků určí, zda snímek obsahuje člověka či ne. U takto ohodnoceného snímku je následně vytvořen soubor, který obsahuje pozitivní či negativní hodnotu na základě klasifikace.



Obrázek 45: Přejechání vedlejších oken z hlavního okna. Náhled vytvořen pomocí nástroje Moqups.

3. Tvorba datasetu

Na základě získaných a klasifikovaných snímků lze vytvořit konkrétní soubor, který reprezentuje datovou sadu. Každý snímek z vybraného adresáře je převeden na stejnou velikost, která je nastavena uživatelem a poté je snímek převeden do odstínů šedi. Výsledný soubor obsahuje sloupce, které představují jednotlivé pixely snímků a poslední sloupec obsahující klasifikační třídy.

4. Normalizace dat

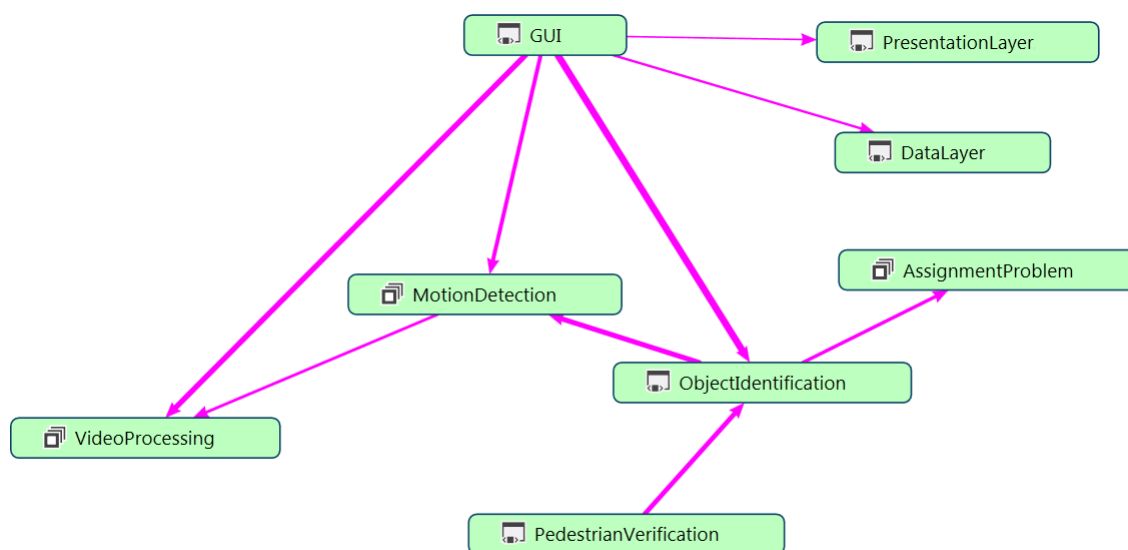
Tato část umožňuje provést normalizaci dat získaných datových sad dle několika metod, které mohou eliminovat extrémní hodnoty v sadách.

5. Konfigurace strojového učení

Poslední část je určena pro konfiguraci vybraných metod strojového učení. Uživatel zde může nastavit parametry těchto metod a poté je natrénovat na vybraných datových sadách.

6.2 Programátorský pohled

Aplikace byla vytvořena pomocí frameworku .NET, knihovny EmguCV a frameworku Accord. NET. Knihovna EmguCV poskytuje vybrané metody pro zpracování obrazu ve výsledné aplikaci. Framework Accord. NET poskytuje vhodnou funkcionalitu pro práci s detektory osob a klasifikačními modely. Uživatelské grafické rozhraní bylo vytvořeno dle knihovny WPF. Aplikace je rozdělena do šesti rozdílných jmenných prostorů podle účelu svého použití následovně:



Obrázek 46: Struktura projektu a vazby mezi jednotlivými jmennými prostory. Vygenerována pomocí nástroje Visual Studio 2015.

- **PedestrianVerification**

Jmenný prostor, který obsahuje třídy a funkcionalitu pro práci s vybranými detektory osob. Součástí jmenného prostoru jsou také třídy sloužící pro vytvoření a manipulaci s datovými sadami a také třídy pro práci s vybranými klasifikačními metodami strojového učení. Hlavní účel této sady je snaha ověřit, zda trajektorie skutečně obsahuje osobu či ne. Nejdůležitější třídou v tomto prostoru je třída `Verification`, která umožňuje pracovat s konkrétním detektorem osob nebo klasifikačním modelem, s kterými je možné provést verifikaci konkrétní trajektorie či celé množiny trajektorií, viz Obrázek č. 55.

- **ObjectIdentification**

Nejdůležitější jmenný prostor, jehož obsahem jsou třídy, které umožňují detekovat a analyzovat trajektorie pohybu osob. Součástí prostoru je hlavním algoritmus, který umožňuje nalézt trajektorie pohybu v celém intervalu videosouboru nebo jen vybrané části. Součástí prostoru jsou třídy, které umí detekovat objekty ze snímku obsahující optický tok. Hlavní třídou je třída `EntriesDetector`,

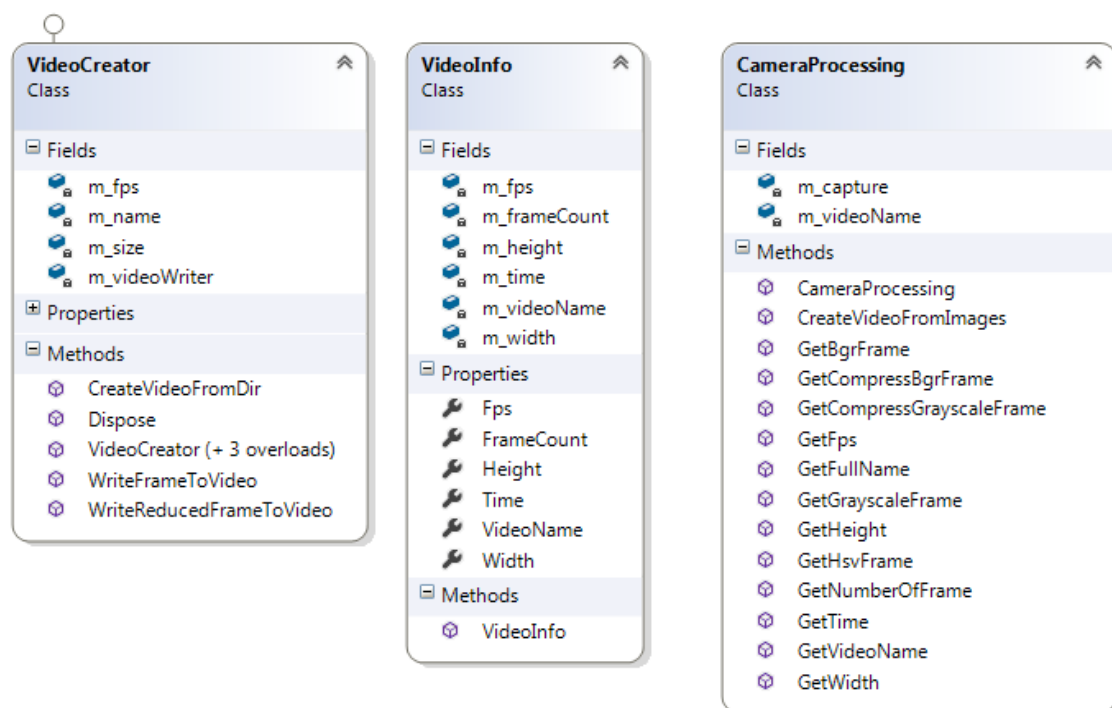
kteřá provádí detekci, verifikaci a analýzu trajektorií. Diagram tříd tohoto prostoru je uveden v příloze, viz Obrázek č. 57.

- **MotionDetection**

Jmenný prostor obsahuje třídy pro práci s vybranými algoritmy pro výpočet optického toku pomocí technologie CUDA či bez ní. Součástí prostoru jsou také třídy, které obsahují funkcionalitu pro různé zobrazení optického toku např. pomocí směrových vektorů nebo v modelu HSV. Stěžejní třídou v tomto jmenném prostoru je třída `OpticalFlowRenderer`, která umí spočítat optický tok pro konkrétní snímek a uložit optický tok dle různého zobrazení. Diagram tříd tohoto prostoru je uveden v příloze, viz Obrázek č. 56.

- **VideoProcessing**

Důležitý jmenný prostor, který je určen pro manipulaci se vstupními videosoubory či množinou snímků a také pro získání požadovaného obrazu z těchto zdrojů v různých barevných modelech a v různé velikosti rozlišení. Prostor také obsahuje třídy pro vytvoření vhodných výstupních videosouborů pomocí různých kompresí obrazu. Celá struktura prostoru je popsána v obrázku č. 47.



Obrázek 47: Diagram tříd, který zobrazuje strukturu projektu pro detekci objektů, vytvořeného nástrojem Visual Studio 2015.

- **GUI**

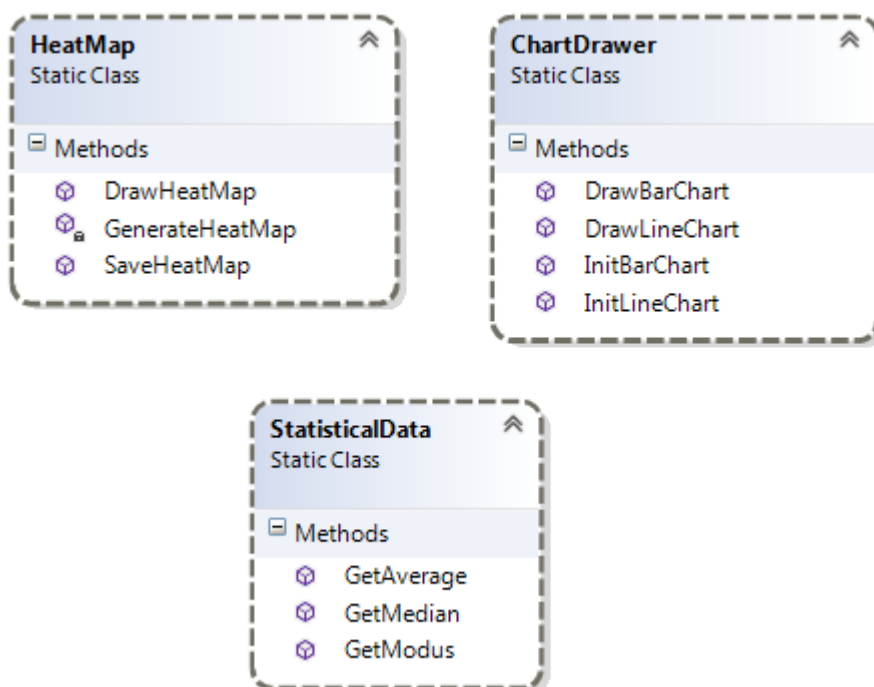
Jmenný prostor využívá knihovnu WPF pro vytvoření vhodného uživatelského grafického rozhraní. Pomocí rozhraní lze poté obsluhovat a nastavovat případné parametry algoritmům, které byly v rámci práce implementovány.

- **AssignmentProblem**

Prostor obsahující funkcionalitu pro vyřešení přiřazovacího problému. Je zde zejména implementována maďarská metoda.

- **PresentationLayer**

Tento prostor je zaměřen na nástroje, které umí vhodně prezentovat získané informace z analyzovaného zdroje. Tyto informace mohou být zobrazeny pomocí teplotních map nebo liniových či sloupcových grafů. Dále je možné vypočítat u získaných informací základní statistické veličiny nebo agregované informace. Struktura prostoru je zobrazena v diagramu tříd, viz Obrázek č. 48.

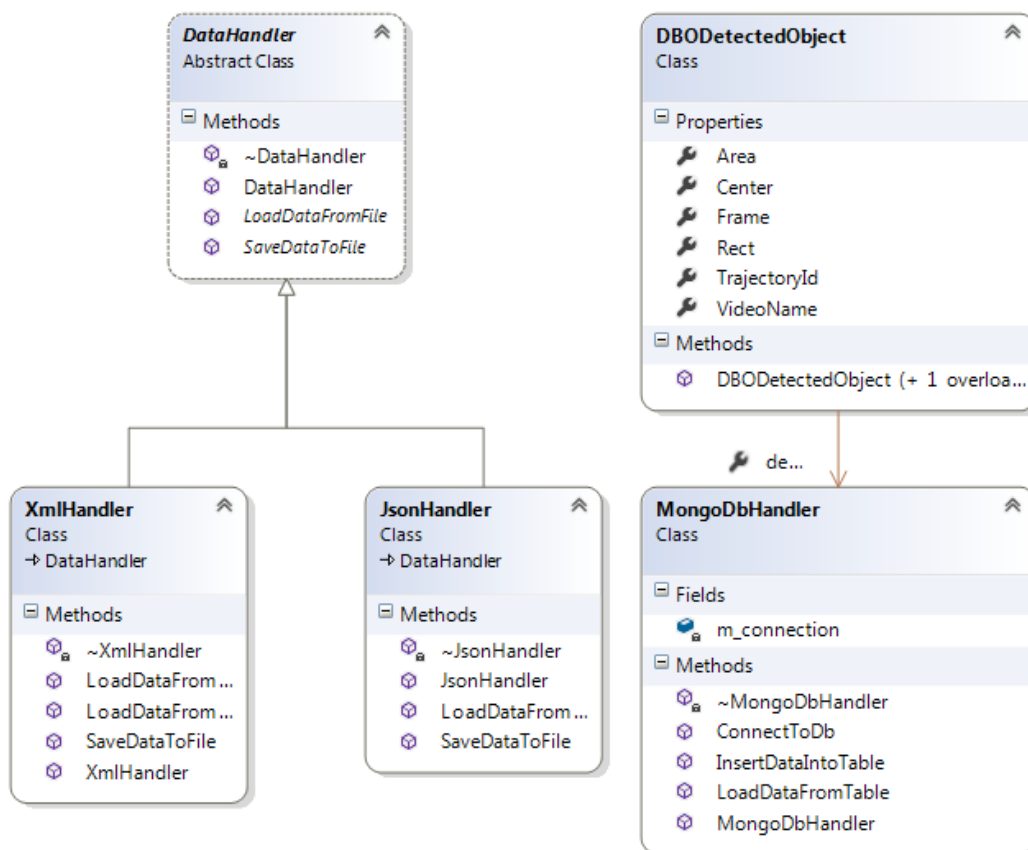


Obrázek 48: Diagram tříd, který zobrazuje strukturu prezentační vrstvy vytvořené nástrojem Enterprise Architect.

Prezentace výsledků může být provedena buď z celého časového intervalu vstupního zdroje nebo na jeho vybrané části. Také mohou být filtrovány a následně zobrazeny jednotlivé trajektorie ze získané množiny informací pomocí výše zmíněných nástrojů.

- **DataLayer**

Jmenný prostor, který obsahuje třídy pro uložení a zpracování nalezených trajektorií pohybu. Trajektorie mohou být uloženy do nerelační databáze MongoDB nebo do formátu XML či JSON. Prostor také obsahuje třídy, které poskytují funkcionalitu pro načítání trajektorií z XML či JSON souborů nebo z databáze MongoDB. Diagram tříd obsahující strukturu prostoru je zobrazen na obrázku č. 49.



Obrázek 49: Diagram tříd, který zobrazuje strukturu datové vrstvy vytvořené nástrojem Enterprise Architect.

7 Dosažené výsledky

Navržená metodika, která byla implementována do komplexní aplikace, byla testována na dvou vybraných videích, viz č. Tabulka 2. Přesnost navržené metodiky byla stanovena na základě komparace reálných a detekovaných trajektorií pohybu a také dle srovnání počtu reálných a vypočítaných odchodů či příchodů do oblasti zachycené kamerou. Dále bylo provedeno srovnání rychlosti zpracování vybraných pořizovaných videí, viz Tabulka č. 10.

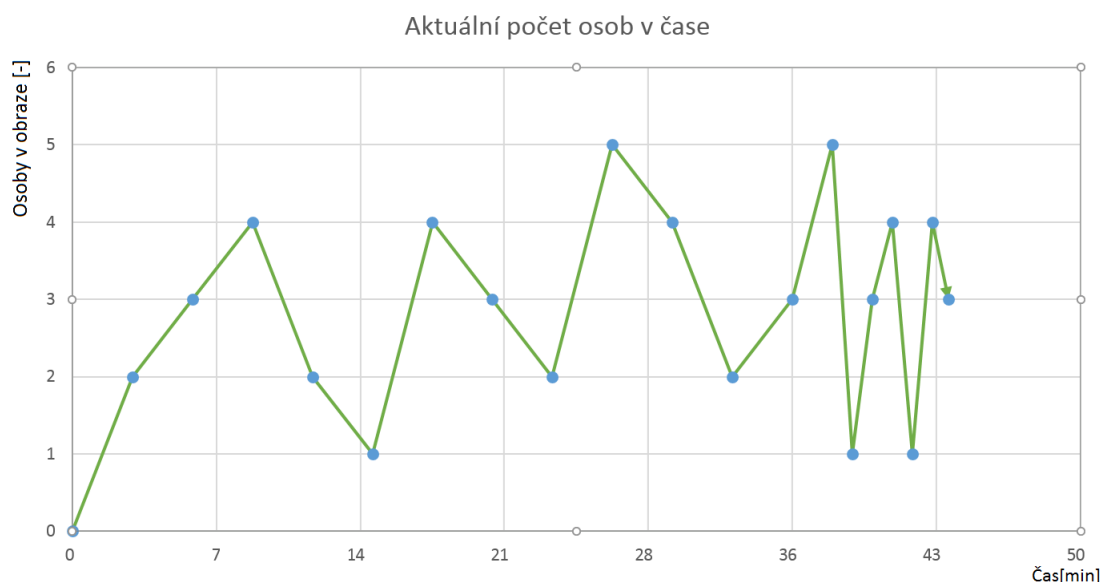
Tabulka 10: Srovnání rychlosti zpracování vstupních videosouborů.

| ID videa | Délka videa | Počet snímků | Doba zpracování |
|----------|-------------|--------------|-----------------|
| 2 | 00:43:48 | 52560 | 05:48:12 |
| 3 | 01:09:05 | 41725 | 09:53:12 |

Dlouhá doba zpracování je výsledkem vysokého rozlišení obou zpracovaných videosouborů a také je doba zpracování ovlivněna použitím TV- L^1 metody.

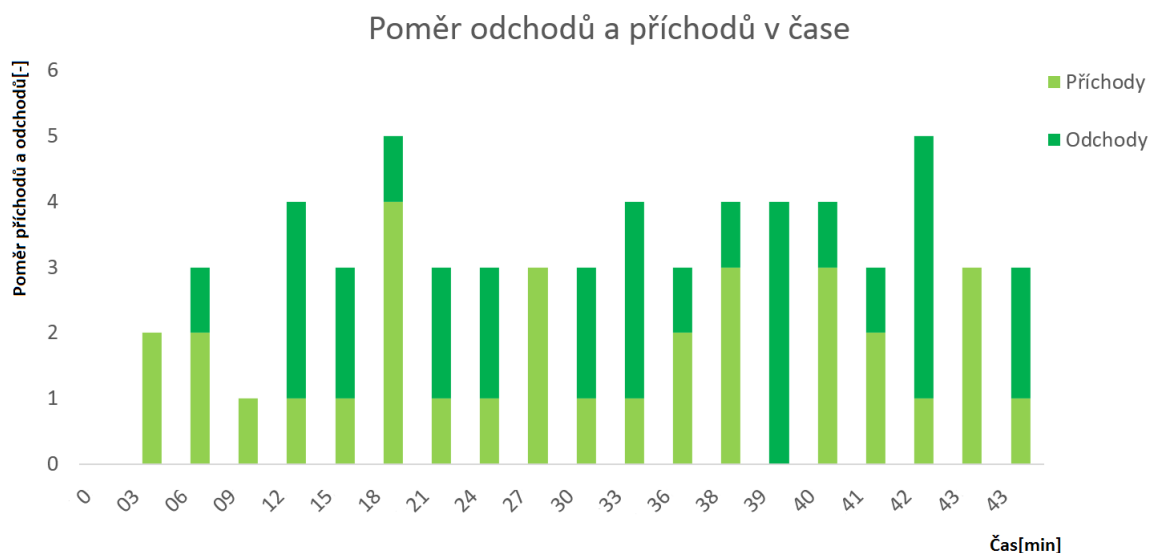
7.1 Testování detekce pohybu osob v prostoru

V rámci prvního testu byla provedena detekce a analýza trajektorií pohybu osob na pořizovaném videu č. 2, viz Tabulka č. 2. Pořizované video obsahuje simulovaný pohyb osob v laboratoři Q11, která reprezentuje kancelářské prostory. Dále byl ze získaných informací vygenerován liniový graf (viz. Obrázek č. 50).



Obrázek 50: Liniový graf, který zobrazuje aktuální počet osob v čase.

Graf zobrazuje aktuální počet osob v obraze v závislosti na čase. Křivka grafu značně kolísá, což je způsobeno častými odchody či příchody do snímané oblasti. Následně byl vytvořen sloupcový graf (viz. Obrázek č. 51), který zachycuje poměr příchodů a odchodů ze snímané oblasti.



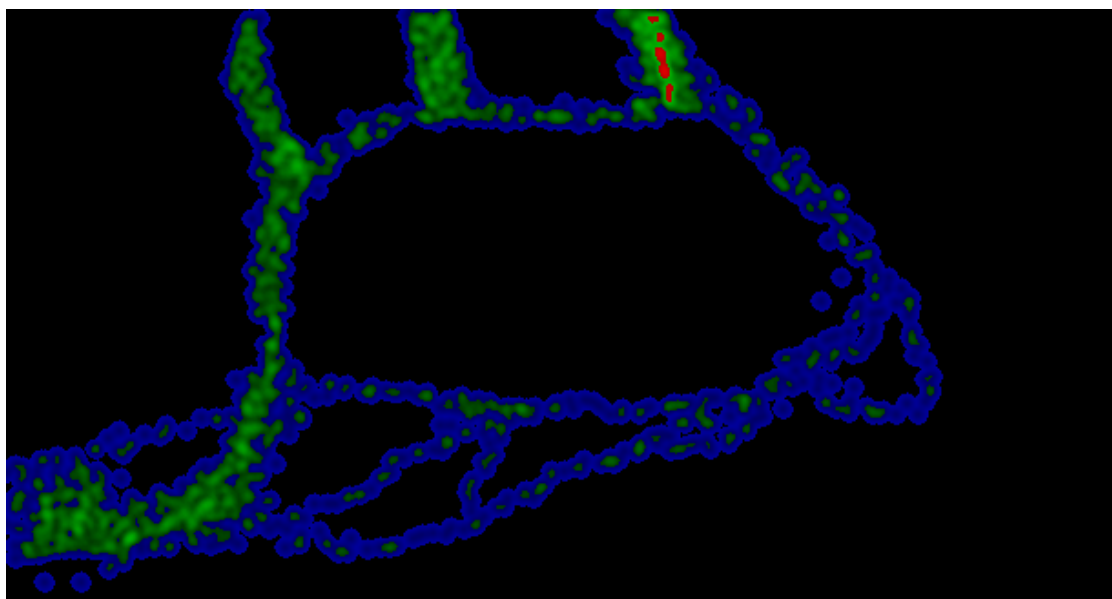
Obrázek 51: Sloupcový graf, který zobrazuje nalezené trajektorie jako poměr příchodů a odchodů osob.

Další údaje, které byly spočítány ze získaných informací, jsou základní statistické charakteristiky, viz Tabulka č. 11. Z charakteristik lze zjistit, že byl průměrný počet osob v obraze 2,92 osob. Nejčastěji se v obraze nacházely čtyři osoby. Dále je možné zjistit, že každé tři minuty průměrně přišlo 1,73 osob do obrazu a průměrně odešlo 1,57 osob z obrazu.

Tabulka 11: Statistické údaje, které byly vypočteny ze získaných informací.

| Typ | Medián | Modus | Průměr pro 3 min. |
|----------------|--------|-------|-------------------|
| Příchody | 1 | 1 | 1,73 |
| Odchody | 1 | 1 | 1,57 |
| Osoby v obraze | 3 | 4 | 2,92 |

Ze získaných informací byla následně vygenerována teplotní mapa, která zobrazuje četnost výskytu osob v čase pomocí různorodých teplot. Z mapy je vidět, že nejčetnější místa se nacházejí ve vstupně-výstupních oblastech snímané oblasti, viz Obrázek č. 54. Z mapy je tak možné vyčíst, že ve spodní části snímané oblasti nebyl zaznamenán frekventovaný pohyb.



Obrázek 52: Ukázka teplotní mapy, která byla vygenerována ze získaných trajektorií pohybu.

Testování je zaměřeno zejména na detekci příchodů a odchodů ze snímaného prostoru. Poté je provedena komparace detekovaných a reálných příchodů či odchodů, podle kterých byla stanovena přesnost algoritmu. Reálné příchody a odchody byly získány dle manuálního zpracování pořízeného videa. Test je také zaměřen na testování jednoduchých případů kolize ke kterým může dojít během výpočtu. Výsledky testu č. 1 jsou uvedeny v tabulce č. 12.

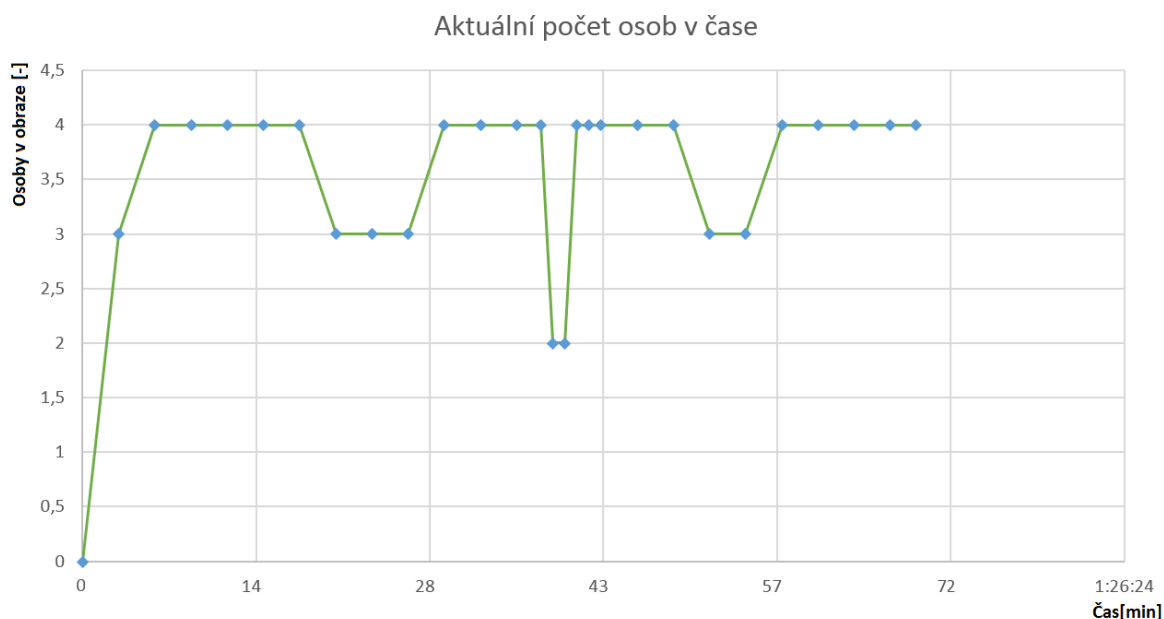
Tabulka 12: Výsledky, které byly získány testováním videosouboru č.2

| | Příchody | Odchody |
|----------|-----------------|----------------|
| Realita | 33 | 30 |
| Detekce | 27 | 25 |
| Přesnost | 81,81% | 83,33% |

7.2 Detekce příchodů a odchodů z místnosti

Druhý test je zaměřen na detekci a analýzu trajektorií pohybu osob na pořízeném videu č. 3 viz tabulka č. 2, proto bylo testování provedeno jiným způsobem než u testu č. 1. V rámci testu byly porovnány skutečné a detekované trajektorie pohybu osob, z čehož byla stanovena přesnost algoritmu. V pořízeném videu je zachycena simulace pohybu osob v laboratoři Q11, která reprezentuje kancelářské prostory. Následně byly spočítány základní statistické charakteristiky ze získaných trajektorií pohybu.

Ze získaných charakteristik je možné zjistit, že průměrně se v obraze nachází 3,5 osob nebo, že se v obraze nejčastěji vyskytují čtyři osoby. Statistické charakteristiky nebyly vypočítány pro příchody a odchody, protože pořízené video neobsahuje téměř žádné případy odchodu či příchodu. Z nalezených informací byl následně vygenerován liniový graf, který zachycuje průběh pohybu osob v čase, viz Obrázek č. 53. Z grafů je patrné, že se téměř neměnil počet osob v obraze, jelikož byl test zaměřen zejména na detekci pohybu osob v obraze.



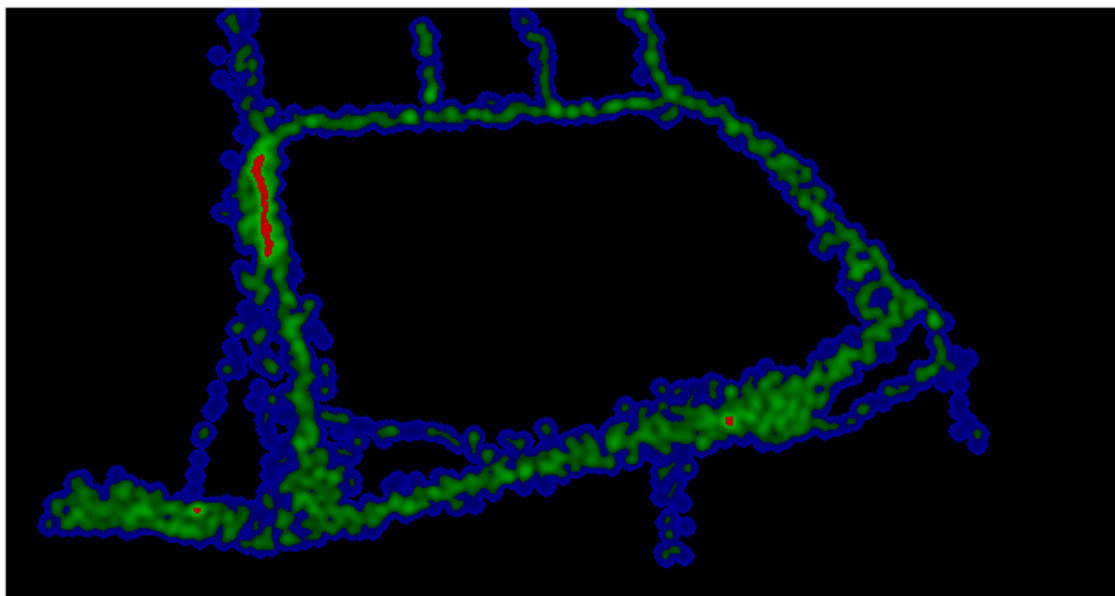
Obrázek 53: Liniový graf, který zobrazuje aktuální počet osob v čase.

Simulace je zaměřena zejména na testování komplexních případů kolize a na složité situace, které mohou nastat během pohybu více osob. Výsledky testu jsou zobrazeny v tabulce č. 13.

Tabulka 13: Výsledky, které byly získány testováním videosouboru č. 1.

| Test | Skutečné trajektorie/Detekované trajektorie | Přesnost |
|------|---|----------|
| 2 | 26/22 | 84,61% |

Dále byla ze získaných informací vytvořena teplotní mapa, která zobrazuje četnost výskytu osob v čase pomocí různorodých teplot, viz Obrázek č. 54. Jelikož byl test zaměřen na testování detekce pohybu osob, tak je z teplotní mapy patrné, že pohyb je rozprostřen rovnoměrně po celé snímané oblasti. Simulace z velké části vypouští příchody a odchody ze snímaného prostoru, což ovlivnilo výslednou teplotní mapu.



Obrázek 54: Ukázka teplotní mapy, která byla vygenerována ze získaných trajektorií pohybu.

Testování navržené metodiky bylo provedeno pomocí dvou komplexních testů, které měly provést testování metodiky na dvou rozdílných simulacích pohybu osob ve snímaném prostoru laboratoře Q11. Každý test byl zaměřen na specifickou část metodiky. Celkové výsledky testování jsou uvedeny v tabulce č. 14. Výsledná přesnost algoritmu byla spočítána aritmetickým průměrem přesnosti dílčích testů. Přesnost byla stanovena na hodnotu 83,25%.

Tabulka 14: Souhrn výsledků, které byly získány testováním pořízených videosouborů.

| Test | Délka videa | Počet snímků | Přesnost |
|------|-------------|--------------|---------------|
| 1 | 00:43:48 | 52560 | 81,81%/83,33% |
| 2 | 01:09:05 | 41725 | 84,61% |

8 Zhodnocení

8.1 Diskuze

Navržená metodika pro detekci a analýzu trajektorií je složena z dílčích částí ovlivňujících výslednou přesnost metodiky, která byla implementována ve frameworku .NET a pomocí dalších výše popsaných knihoven. Prvním faktorem, který ovlivňuje přesnost algoritmu je zvolený algoritmus optického toku. Tyto algoritmy byly porovnány ve čtvrté kapitole. Na základě provedené komparace byl zvolen algoritmus TV-L1, který vykazoval nejvyšší přesnost detekce tj. 3% Miss rate při nejvyšším rozlišení za cenu snížené rychlosti výpočtu. V případě, že by byl kladen důraz na rychlejší zpracování vstupního zdroje za cenu horší přesnosti, tak je možné aplikovat metodu Lucas-Kanade pro detekci dynamických objektů.

Následně bylo provedeno srovnání vybraných metod pro verifikaci trajektorií, které byly natrénovány a optimalizovány dle datové sady č. 4, která obsahovala nejrepresentativnější vzorky ze všech datových sad. Nejlepší metodou pro verifikaci trajektorií byla zvolena vícevrstvá neuronová síť a není tedy doporučeno používat ostatní metody, které nedosahovali takových výsledků. Neuronovou síť lze dále optimalizovat použitím rozsáhlejší datové sady.

Výše zmíněné metody zásadně ovlivňují navrženou metodiku. Pro verifikaci navržené metodiky byly provedeny dva rozdílné komplexní testy, které testovali zásadní části metodiky. První test byl zaměřen na detekci příchodů či odchodů a stanovil výslednou přesnost algoritmu na 83,33% pro detekci odchodů z oblasti, respektive 81,81% pro detekci příchodů do snímané oblasti. Ručním zpracováním bylo nalezeno v pořízeném videu 33 příchodů respektive 30 odchodů. Algoritmus celkově detekoval 27 příchodů a 25 odchodů. Zásadním faktorem, který ovlivňuje výslednou přesnost, je volba velikosti a pozice pomocných oblastí, které definují vstupně-výstupní prostory.

Druhý test byl zaměřen na testování pohybu osob ve složitějších situacích. V testu byly porovnány reálné trajektorie, které byly nalezeny ručním zpracováním v pořízeném videosouboru a detekované trajektorie získané pomocí vytvořené aplikace. Celkově bylo nalezeno 26 skutečných trajektorií. Aplikace detekovala 22 trajektorií v daném videosouboru. Výsledná přesnost algoritmu byla tedy stanovena na 84,61%. Přesnost detekce trajektorií byla ovlivněna nesprávným spárováním nalezených objektů do správné trajektorie či nevhodnou predikcí polohy Kalmanovým filtrem. Výsledná přesnost algoritmu byla stanovena pomocí aritmetického průměru na 83,25% pro dané zdroje dat.

Vytvořený software může být využit pro komplexnější pohled na chování osob ve snímané ploše v mnoha oblastech. Software může sloužit pro podporu manažerského rozhodování či analýzu chování zákazníka. Manažer může zjistit pomocí vygenerovaných teplotních map zájem zákazníků o určitou skupinu produktů v dané sekci obchodu apod. Software může dále sloužit pro monitoring zaměstnanců ve snímané pracovní oblasti.

8.2 Shrnutí

Hlavním cílem práce bylo navrhnout metodiku, která dokáže detekovat a analyzovat trajektorie pohybu osob v dynamickém obraze. Následně byla tato metodika implementována do komplexního software s pomocí frameworku .NET a dalších podpůrných knihoven. Práce je rozdělena do několika kapitol, které popisují postup řešeného problému.

První kapitola definuje cíl práce a její úvod. Následující kapitola, která je zaměřena na rešeršní část práce, popisuje vybrané metody pro detekci dynamických objektů v obraze, metody pro detekci osob v obraze, a také je popsán princip zpracování obrazu z kterého vychází praktická část práce.

Třetí kapitola popisuje vlastnosti pořízených videosouborů, které zachycují simulaci pohybu osob ve snímaném prostoru. Videosoubory sloužily pro testování navržené metodiky a zejména pro vytvoření datových sad. Datové sady byly následně převedeny do stejného formátu a sloužily pro výběr optimální metody pro verifikaci trajektorií na základě testovacího procesu.

Další kapitola je zaměřena na komparaci vybraných metod podle vybraných kritérií pro detekci dynamických objektů dle vybraných algoritmů pro výpočet optického toku. Dále bylo provedeno srovnání metod, které jsou schopny klasifikovat trajektorie pohybu osob do vybraných tříd nebo metod, které dokážou detekovat přibližnou polohu osob v obraze.

Pátá kapitola je zásadní, protože definuje metodiku pro řešení zadaného problému. V kapitole je popsána metodika pro detekci a analýzu trajektorií pohybu podle vybraných metod, které byly stanoveny dle výše zmíněné komparace. Metodika je rozdělena do dvou částí dle svého účelu. První část metodiky popisuje detekci trajektorií, které jsou reprezentovány objekty detekované dle algoritmu $TV-L^1$ a jsou poté následně spárovány maďarskou metodou či je jejich poloha predikována Kalmanovým filtrem.

Druhá část metodiky je zaměřena na návrh řešení, zda výše detekované trajektorie patří lidem či ne. Následujícím krokem druhé části metodiky je analýza vlastností trajektorií. Následující kapitola je zaměřena na implementaci navržené metodiky do komplexního software pomocí frameworku .NET a dalších nástrojů. Kapitola popisuje dva pohledy z hlediska uživatele a programátorského návrhu. V kapitole je popsán základní princip ovládání aplikace, optimální parametry aplikace a grafické rozhraní aplikace. Dále jsou popsány nástroje, které slouží pro zobrazení získaných výsledků a také pro zpracování získaných informací. Programátorský návrh popisuje implementaci metodiky a samotnou strukturu projektu. Dále jsou popsány dílčí prvky projektu a jejich funkcionalita dle zvolených nástrojů.

Poslední kapitola popisuje verifikaci navržené metodiky podle dvou komplexních testů, které testují vybrané části navržené metodiky a vyhodnocují výslednou přesnost metodiky.

8.3 Závěr

Hlavním cílem práce byl návrh metodiky pro detekci a analýzu trajektorií pohybu osob ve vytyčeném prostoru snímaného průmyslovou kamerou dle algoritmů optického toku. Souvisejícím cílem byla implementace navržené metodiky do komplexního software pomocí frameworku .NET. Navržená metodika byla testována pomocí komplexních testů, které otestovali přesnost navržené metodiky. Výsledná přesnost je stanovena na 83,25% pro testované zdroje dat. Vytvořený software je nalézt trajektorie pohybu osob z různorodých datových zdrojů, ve kterých se vyskytuje pohyb osob. Získané trajektorie lze poté zobrazit dle různých nástrojů pro zajímavější pohled na data. Z daného výsledku vyplývá, že cíl práce byl splněn.

9 Literatura

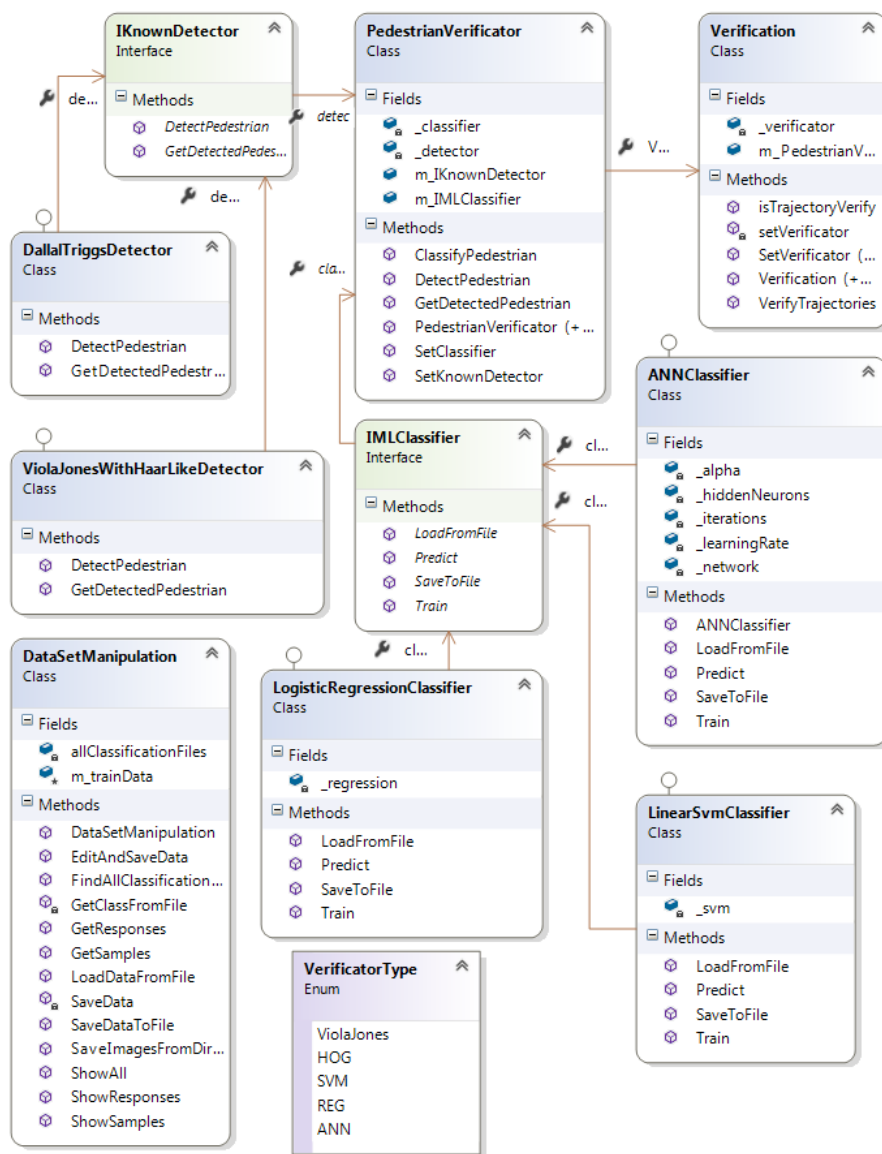
- BARRON, J.L., THACKER, N.A. *Tutorial: Computing 2D and 3D Optical Flow [online]*. Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester, 2005 [cit. 2017-01-27]. Dostupné z: <http://tina.wiau.man.ac.uk/docs/memos/2004-012.pdf>.
- DALLAL, N., TRIGGS, B. *Histograms of Oriented Gradients for Human Detection*. INRIA Rh à one-Alps, 655 avenue de l'Europe, Montbonnot 38334, France. 2005.
- DE BOER, J., KALKSMA, M. *Choosing between optical flow algorithms for UAV position change measurement [online]*. 2015. [cit. 2017-01-27]. Dostupné z: http://www.nhlcomputervision.nl/wp-content/uploads/SC2015_opticalflow.pdf.
- FAWCETT, T. *ROC Graphs: Notes and Practical Considerations for Researchers*. HP Laboratories: Kluwer Academic Publishers, 2004.
- FRIEBELOVÁ, J. *Rozhodovací modely v praxi: DISTRIBUČNÍ ÚLOHY [online]*. 2006 [cit. 2017-03-29]. Dostupné z: http://www2.ef.jcu.cz/~jfried/rmp/teorie_oa.php.
- HÁJOVSKÝ, R., PUSTKOVÁ R., KUTÁLEK F. *ZPRACOVÁNÍ OBRAZU V MĚŘICÍ A ŘÍDICÍ TECHNICE: učební text [online]*. Vysoká škola báňská – Technická univerzita Ostrava, 2012 [cit. 2017-03-29]. Dostupné z: http://www.person.vsb.cz/archivcd/FEI/ZOMRT/Zpracovani_obrazu_v_merici_a_ridici_technice.pdf.
- HAVLENA, V., ŠTECHA J. *Moderní teorie řízení*. 1999. České vysoké učení technické v Praze, Fakulta elektrotechnická: Ediční středisko CVUT, Praha 6, Žitná 4, 1999.
- HORÁK, K. *Zpracování vícerozměrných signálů [online]*. Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, Vysoké učení technické v Brně, 2010 [cit. 2017-03-28]. Dostupné z: http://midas.uamt.feec.vutbr.cz/ZVS/zvs_cz.php.
- HORÁK, K., KALOVÁ I., PETYOVSÝ P., RICHTER M. *Počítačové vidění [online]*. 2008. Fakulta elektrotechniky a komunikačních technologií, Vysoké učení technické v Brně, 2008 [cit. 2017-03-28]. Dostupné z: http://www.uamtold.feec.vutbr.cz/vision/TEACHING/MPOV/Pocitacove_videni_S.pdf.
- HUSTON, STEPHEN J. A HOLGER G. KRAPP. *Visuomotor transformation in the fly gaze stabilization system*. 2008. P1 o S Biology, v.6, no.7, 2008 July, p.1468: Public Library of Science. ISSN 1544-9173.

- Introduction - Accord.NET Machine Learning in C#: The Accord.NET Image Processing and Machine Learning Framework* [online]. 2015 [cit. 2017-05-17]. Dostupné z: <http://accord-framework.net/intro.html>.
- KAEHLER, A., BRADSKI, G. *Learning OpenCV*. 1005 Gravenstein Highway North, Sebastopol: O'Reilly Media, 2008. ISBN 978-0-596-51613-0.
- KLOS, D. *Počítání tlakových lahví v obraze* [online]. Diplomová práce. Vysoké učení technické v Brně, fakulta informačních technologií, ústav počítačové grafiky a multimédií. 2014 [cit. 2017-02-12]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=119345.
- KRÁLÍK, M. *Detekce objektů v obraze s pomocí rozšířené sady Haarových příznaků a histogramu* [online]. 2012 [cit. 2017-02-17]. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=53830. Diplomová. VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací.
- MÄENPÄÄ, T. *THE LOCAL BINARY PATTERN APPROACH TO TEXTURE ANALYSIS: EXTENSIONS AND APPLICATIONS* [online]. Infotech Oulu and Department of Electrical and Information Engineering, University of Oulu, 2003 [cit. 2017-05-17]. Dostupné z: <http://jultika.oulu.fi/files/isbn9514270762.pdf>.
- OpenCV 2.4.13.2 documentation: Introduction to Support Vector Machines* [online]. 2014 [cit. 2017-05-17]. Dostupné z: http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html.
- OpenCV 3.0.0-dev documentation: Background Subtraction* [online]. 2014 [cit. 2017-04-16]. Dostupné z: http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_video/py_bg_subtraction/py_bg_subtraction.html.
- ORSÁK, F. *Algoritmus sledování objektů pro funkční vzor sledovacího systému*. Technická zpráva - FIT - VG20102015006 - 2014 - 04 [online]. Fakulta informačních technologií, Vysoké učení technické v Brně, 2004 [cit. 2017-05-17]. Dostupné z: http://www.fit.vutbr.cz/research/pubs/tr.php?file=%2Fpub%2F11033%2F04_forsag_tracking.pdf&id=11033.
- PICCARDI, M. *Background subtraction techniques: a review* [online]. Faculty of Information Technology University of Technology, Sydney (UTS), Australia, 2004 [cit. 2017-02-22]. Dostupné z: http://profs.sci.univr.it/~cristanm/teaching/sar_files/lezione4/Piccardi.pdf.
- SÁNCHEZ, J., MEINHARDT-LLOPIS, E., FACCIOLO, G. *TV-L1 Optical Flow Estimation* [online]. Image Processing On Line on 2013, 2013 [cit. 2017-02-16]. Dostupné z: <http://www.ipol.im/pub/art/2013/26/article.pdf>.

- SCHAPIRE, R.E. *Explaining adaboost*. [online]. In: Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik, 2013 [cit. 2017-03-27]. Dostupné z: <http://rob.schapiire.net/papers/explaining-adaboost.pdf>.
- SHI, SHIN. *Emgu CV essentials: develop your own computer vision application using the power of Emgu CV*. Birmingham, UK: Packt Publishing, 2013. ISBN 978-1-78355-952-7.
- StatSoft: Úvod do neuronových sítí* [online]. 2013 [cit. 2017-03-27]. Dostupné z: http://www.statsoft.cz/file1/PDF/newsletter/2013_02_05_StatSoft_Neuronove_site_linky.pdf.
- VIOLA, P., JONES, M. *Rapid object detection using a boosted cascade of simple features*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2001, pp. I511-I518. 2001.
- WEDEL, A., POCK, T., ZACH, C., BISCHOF, H., CREMERS, D. *An improved algorithm for TV-L1 optical flow*. doi:10.1007/978-3-642-03061-12. 2009.

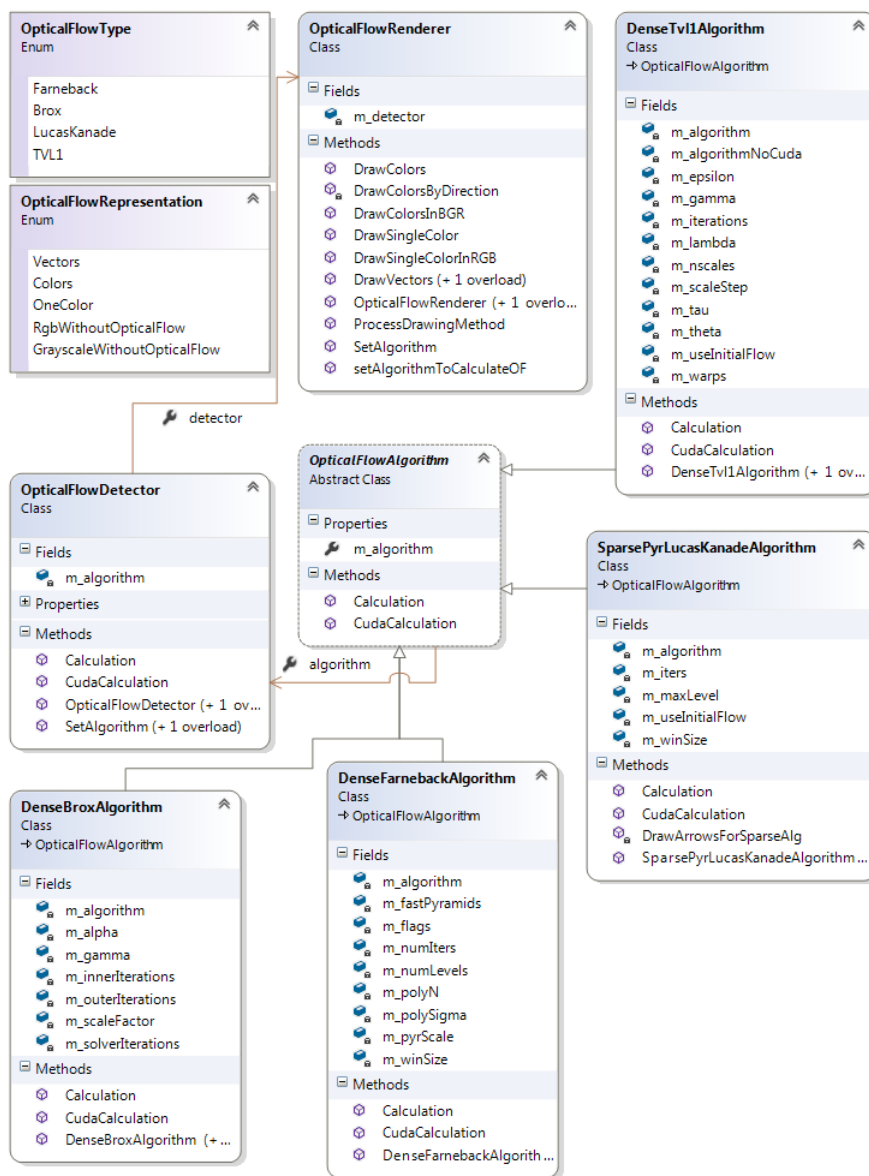
Přílohy

A Diagram tříd jmenného prostoru PedestrianVerification



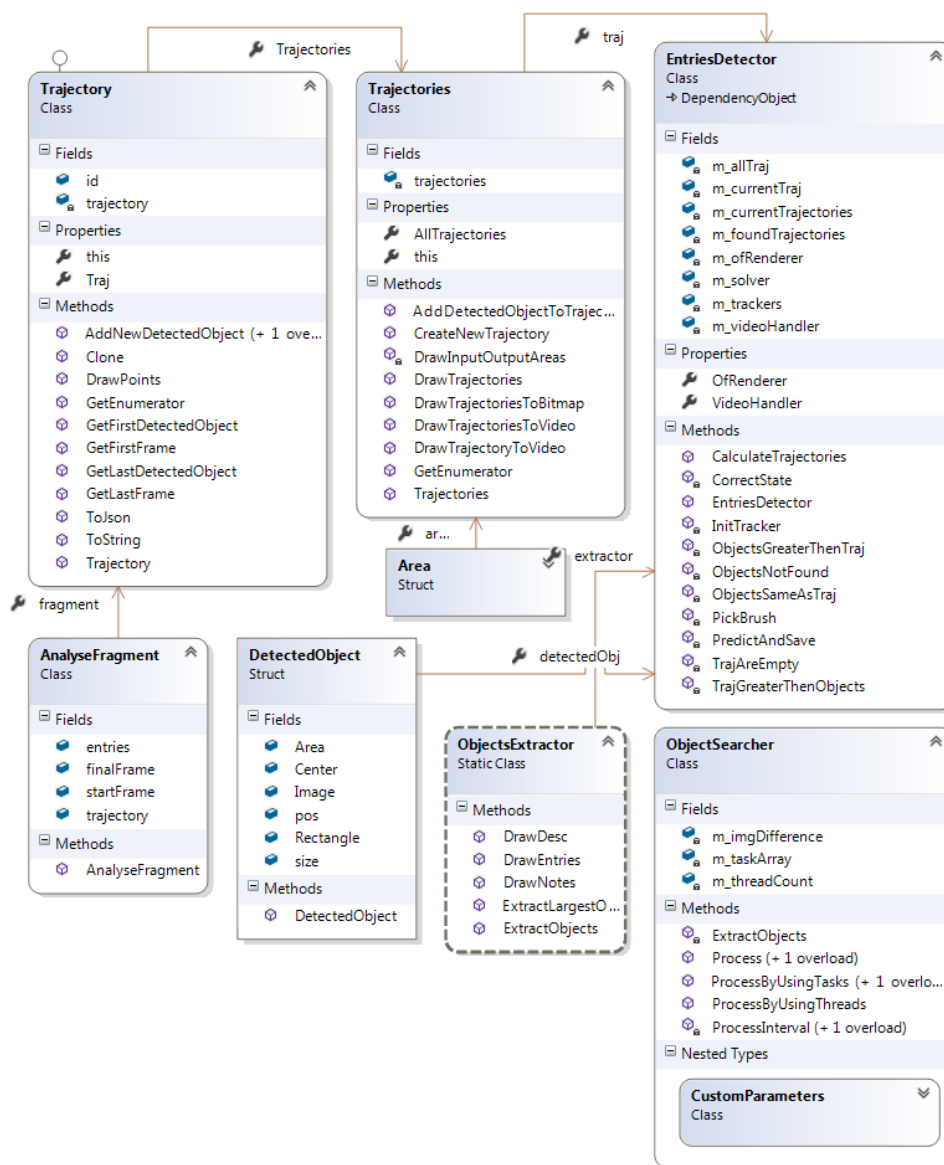
Obrázek 55: Diagram tříd, který zobrazuje strukturu projektu verifikace, vytvořeného nástrojem Visual Studio 2015.

B Diagram tříd jmenného prostoru MotionDetection



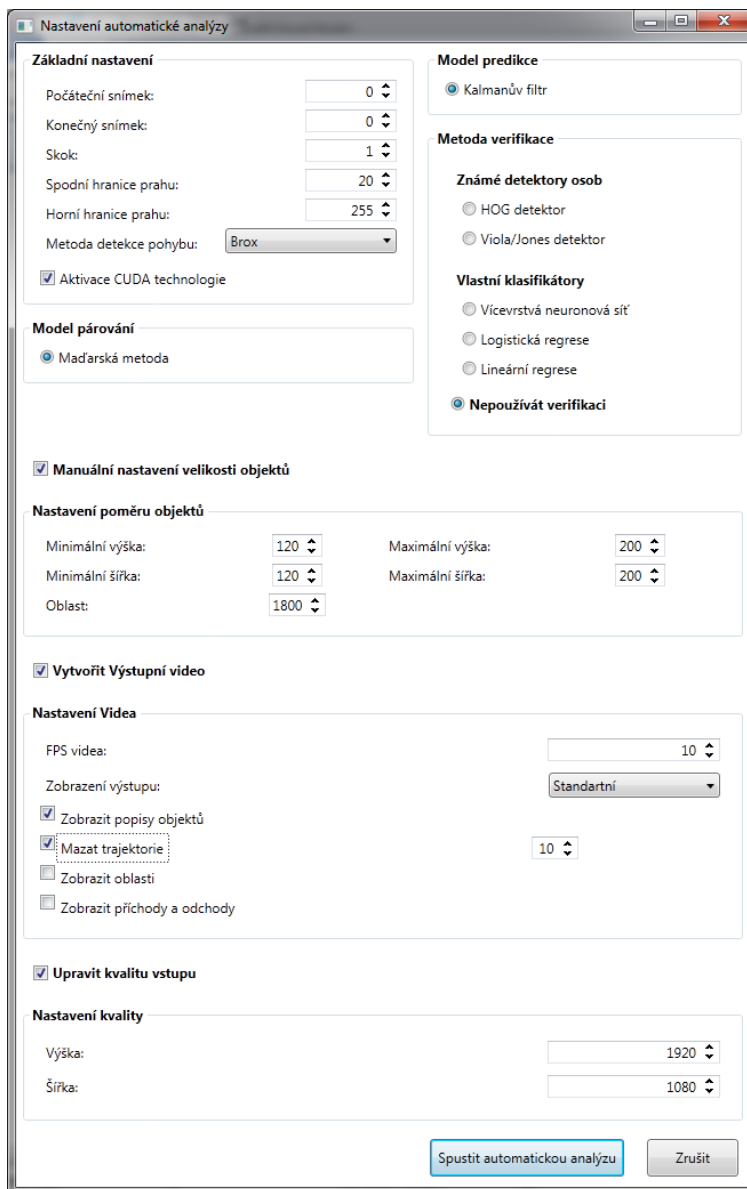
Obrázek 56: Diagram tříd, který zobrazuje strukturu projektu pro detekci pohybu, vytvořeného nástrojem Visual Studio 2015.

C Diagram tříd jmenného prostoru ObjectIdentification



Obrázek 57: Diagram tříd, který zobrazuje strukturu projektu pro detekci objektů, vytvořeného nástrojem Visual Studio 2015.

D Grafické rozhraní pro nastavení algoritmu



Obrázek 58: Grafické uživatelské rozhraní, které bylo vytvořeno pomocí WPF. GUI slouží pro nastavení parametrů algoritmu.