

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

# **Analýza řidiče za pomoci sférických kamer**

## **Driver Analysis Using Spherical Cameras**

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne. Uviedol som všetky literárne  
pramene a publikácie, z ktorých som čerpal.

V Ostrave 1. apríla 2020

.....

Súhlasím so zverejnením tejto diplomovej práce podľa požiadaviek čl . 26, odst. 9 Študijného a skúšobného poriadku pre štúdium v bakalárskych programoch VŠB-TU Ostrava.

V Ostrave 1. apríla 2020

.....

Rád by som poďakoval môjmu vedúcemu práce Ing. Radovanovi Fusekovi za pomoc a ochotu pri vypracovaní diplomovej práce

## **Abstrakt**

Hlavnou témou diplomovej práce je rozpoznávanie a analýza vodiča v aute pomocou sférických kamier. Táto práca je rozdelená do viacerých samostatných častí. Prvá časť spočíva v samotnej detekcii ľudí a ich aktivít sférickou kamerou, hľadanie nedostatkov a nájdenie optimálnych parametrov pre čo najefektívnejšiu detekciu. Druhá časť je zameraná na porovnanie jednotlivých knižníc a metód, ktoré sa používajú na analýzu ľudského tela a tváre v obraze. Posledná časť je venovaná porovnaniu týchto metód s použitím reálnych dát zachytených sférickou kamerou a zhrnutie výsledkov.

**Kľúčové slová:** Sférická kamera, detekcia obrazu, analýza ľudskej tváre, detekcia ľudí, vodič

## **Abstract**

Main focus of this Diploma thesis is detection and analysis of driver in car with help of spherical cameras. This thesis is divided into few parts. The first part is about detection itself, detection of people by spherical cameras, research of disadvantages and finding optimal parameters for most efficient detection. The second part is focused on comparison of libraries used for human body and face detections. The last part is about comparison of libraries with real data captured by spherical camera and summary of results.

**Keywords:** Spherical camera, image detection, analysis of human face, pedestrian detection, driver

# Obsah

<b>Zoznam použitých skratiek a symbolov</b>	<b>7</b>
<b>Zoznam obrázkov</b>	<b>8</b>
<b>Zoznam výpisov zdrojového kódu</b>	<b>9</b>
<b>1 Úvod</b>	<b>10</b>
<b>2 Detekcia a analýza ľudského tela v obrazoch</b>	<b>11</b>
2.1 Haar . . . . .	11
2.2 HOG . . . . .	13
2.3 OpenPose . . . . .	15
2.4 TensorFlow . . . . .	18
2.5 Ostatné metódy . . . . .	19
<b>3 Detekcia vodiča vo vozidle</b>	<b>20</b>
<b>4 Využitie sférických kamier na detekciu obrazu</b>	<b>21</b>
4.1 Použitie v analýze videa . . . . .	22
4.2 Technické parametre . . . . .	22
<b>5 Program</b>	<b>23</b>
5.1 Požiadavky a návrh programu . . . . .	24
5.2 Detekcia vodiča . . . . .	25
5.3 Orientácia hlavy . . . . .	27
5.4 Výstup programu . . . . .	28
5.5 Porovnanie výsledkov . . . . .	29
5.6 Využitie zozbieraných dát . . . . .	30
5.7 Používateľská príručka . . . . .	31
<b>6 Možnosti vylepšenia detekcie</b>	<b>32</b>
<b>7 Záver</b>	<b>33</b>
<b>Literatúra</b>	<b>34</b>

## Zoznam použitých skratiek a symbolov

2D	– 2-dimensional
3D	– 3-dimensional
CNN	– Convolutional neural network
CPU	– Central processing unit
FPS	– Frames per second
GPU	– Graphical processing unit
HOG	– Histogram oriented gradients
IR	– Infra Red
LED	– Light Emitting Diode
OpenCV	– Open Source Computer Vision
PAF	– Part Afinity Fields
PX	– Pixel

## Zoznam obrázkov

1	Výpočet integrálneho obrazu - vstupný obraz (A) , integrálny obraz (B) . . . . .	11
2	Haar - dvoj-obdĺžnikové príznaky (A, B), troj-obdĺžnikové príznaky (C) a štvor- obdĺžnikové príznaky(D) [1] . . . . .	12
3	Haar - detekcia tváre v slabých svetelných podmienkach . . . . .	13
4	HOG - séria krokov [2] . . . . .	14
5	HOG - vstupný obraz (a), normalizácia gradientu (b), orientácia gradientu (c), rozdelenie do buniek (d) vypočítané histogramy (e). [3] . . . . .	14
6	OpenPose - odhad viacerých ôsob v reálnom čase pomocou polí afinity filtra. [4] .	15
7	COCO - označenie častí tela v COCO datase. [5] . . . . .	16
8	BMW driving assistant - umiestnenie kamery na snímanie vodiča. [6] . . . . .	20



## Zoznam výpisov zdrojového kódu

1	Množina párov končatín v datasete COCO . . . . .	17
---	--	----

# 1 Úvod

V dnešnom modernom svete sú autá takmer každodennou súčasťou života ľudí. Mnohokrát sa ani nezamýšľame nad ich bezpečnosťou, ktorá je v prípade zrážky kľúčová. V súčasnosti nám pri jazde autom asistuje veľké množstvo systémov, ktoré zvyšujú bezpečnosť posádky, ale aj ostantých účastníkov cestnej premávky. Aj keď tieto systémy ešte stále nedokážu vodiča úplne nahradiť, dokážu mu výrazným spôsobom pomôcť napríklad v krízových situáciách. Výhodou takýchto systémov je ich rýchlejší reakčný čas oproti človeku. Takéto systémy spočívajú v použití rôznych snímačov alebo kamier, ktoré aktívne sledujú okolie ale aj interiér vozidla. Vďaka takýmto moderným technickým riešeniam je možné predísť rôznym častokrát aj smrteľným dopravným nehodám. Výrobcovia áut sa čoraz častejšie snažia svoje systémy vylepšovať na čo najvyššiu možnú úroveň a poskytnúť tak vysoký level ochrany.

Táto diplomová práca sa zameriava hlavne na problematiku analýzy vodiča pomocou detekcie obrazu zo sférickej (360-stupňovej) kamery. V diplomovej práci som sa venoval analýze videa z kamery umiestnenej v interiéri vozidla. Vhodným umiestnením kamery je možné získať obraz zpred auta, ale aj obraz vodiča sediaceho za volantom. V tejto práci som sa zameriaval na analýzu a spracovanie videa z interiéru vozidla na zachytenie ľudských aktivít vodiča. Aby som získal čo najväčšiu časť tela vodiča, je potrebné mať dostatočne veľký uhol záberu. Bežné kamery majú uhol záberu veľmi nízky, aby dokázal z malej vzdialenosti zachytiť celý snímaný objekt. Takýto problém sa naskytuje najpríklad aj v interiéri vozidla, kde je vzdialenosť kamery od snímaného objektu menej ako 1 meter, čo nemusí byť dostatočné na zosnímanie tela celého vodiča. Práve v takejto situácii je vhodné použiť širokouhlú prípadne sféricкую kameru. Počas práce som mal k dispozícii viaceré kamery, s ktorými som zhotovil niekoľko desiatok videí v rôznych situáciách. Z takýchto videí som dokázal analyzovať a zistiť mnoho užitočných informácií, ktore sú spracované v tejto diplomovej práci. Tieto informácie som zbieral nahrávaním videa sférickými kamerami za rôznych svetelných podmienok a pozícií vodiča. V tejto práci sú taktiež spomenuté problémy takejto analýzy, riešenia vzniknutých problémov, ale aj zhrnutie celkovej problematiky sledovania vodiča vo vozidle. V práci sú tiež zhrnuté ďalšie možnosti vylepšenia detekcie a porovnanie oproti klasickým kamerám.

V nasledujúcich kapitolách je postupne rozobratá problematika snímania ľudských postáv v obrazoch, a skúmanie ich aktivít. Pre snímanie postavy som sa rozhodol použiť viacero metód, ktoré som následne porovnal a zanalyzoval. Aby som vedel vyhodnotiť správnu pozíciu vodiča, rozhodlo sa použiť neurónovú sieť, ktorú som trénoval na vlastnom datasete.

V súčasnosti som taktiež nenašiel veľa riešení na spracovanie videa zo sférickej kamery a preto by som sa snažil zamerať túto prácu hlavne na túto oblasť. Pri analýze vodiča som taktiež nenašiel vhodné datasety z interiéru vozidla snímané sféricickou kamerou.

## 2 Detekcia a analýza ľudského tela v obrazoch

História detekcie postáv v obrazoch siaha až do polovice 20. storočia. Mnoho inžinierov videlo obrovský potenciál detekcie obrazu napríklad v oblastiach medicíny, priemyslu, dopravy a mnohých ďalších oblastiach. S nárastom technických možností postupne rástla aj motivácia využiť detekciu obrazu aj v praxi. Jeden z prvých vedeckých článkov v oblasti spracovania obrazu [7] rozoberal napríklad jednoduchú analýzu obrazu a spracovanie obrazov s dostupnými prostriedkami. Postupom času sa však počítačová technika vylepšovala a bolo možné pracovať na vývoji metód pre analýzu a detekciu objektov v obrazoch. Na detekciu chodcov alebo iných ľubovoľných objektov existuje mnoho prístupov. Veľkým fenoménom v posledných rokoch sa stali neurónové siete. Okrem neurónových sietí však stále existujú aj tradičné metódy, ktoré fungujú aj bez tréningových dát. Vo svojej práci som pracoval hlavne s metódami Haar a HOG, ktoré sa radia medzi najpoužívanéjšie tradičné metódy a sú im venované samostatné podkapitoly 2.1 a 2.2.

Každý obraz sa skladá z pixelov. Analýza obrazu však nespočíva v prehľadávaní jednotlivých pixelov, ale v hľadaní jednotlivých objektov v obraze. Tieto objekty je možné určovať do samostatných tried. Triedy nám určujú, aký druh objektu sa v obraze nachádza (Napríklad chodec, vozidlo, dopravná značka a podobne). Aby bolo možné tieto objekty (v našom prípade ľudí) nájsť, bolo potrebné nájsť spoľahlivý a rýchly spôsob detekcie.

### 2.1 Haar

Táto metóda bola popísaná autormi Viola a Jones [8] v roku 2001. Medzi jej hlavné výhody patrí vysoká rýchlosť a spoľahlivá detekcia a vysoká nezávislosť na intenzite osvetlenia. Vo všeobecnosti je tento detektor rozdelený do 4 samostatných častí: Výpočet integrálneho obrazu, výpočet Haar príznakov, výber príznakov a kaskádový klasifikátor.

Výpočet integrálneho obrazu sa robí prevedením vstupného obrazu na integrálny obraz (Obr. 1). Výpočet pre konkrétne súradnice  $(x, y)$  spočíva v súčte hodnôt jasů vľavo a nad súradnicami  $(x, y)$ . Výpočet je znázornený v rovnici 1.

1	1	1
1	1	1
1	1	1

**(A)**

1	2	3
2	4	6
3	6	9

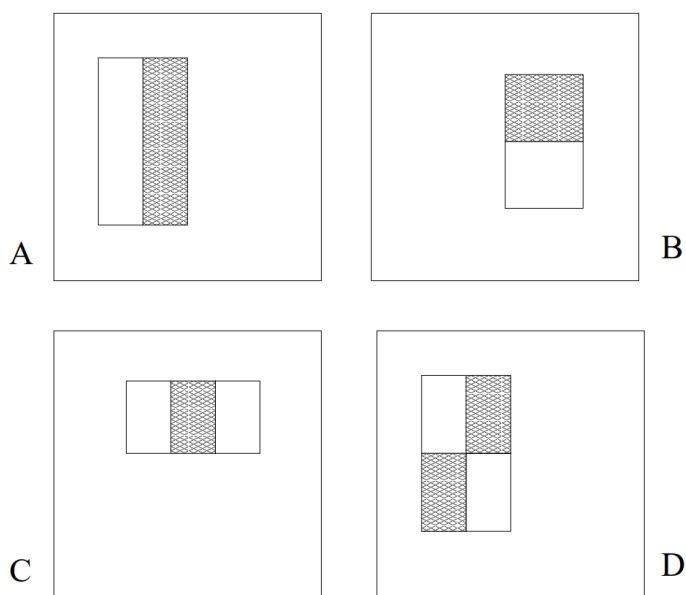
**(B)**

Obr. 1: Výpočet integrálneho obrazu - vstupný obraz (A) , integrálny obraz (B)

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \quad (1)$$

Metóda funguje na porovnaní celých blokov pixelov. Tieto bloky (často nazývané aj zhluky) môžu mať rôzne tvary, veľkosť a natočenie. Tieto bloky môžu nadobúdať rôzne tvary ale vo všeobecnosti sa používajú 3 hlavné typy príznakov:

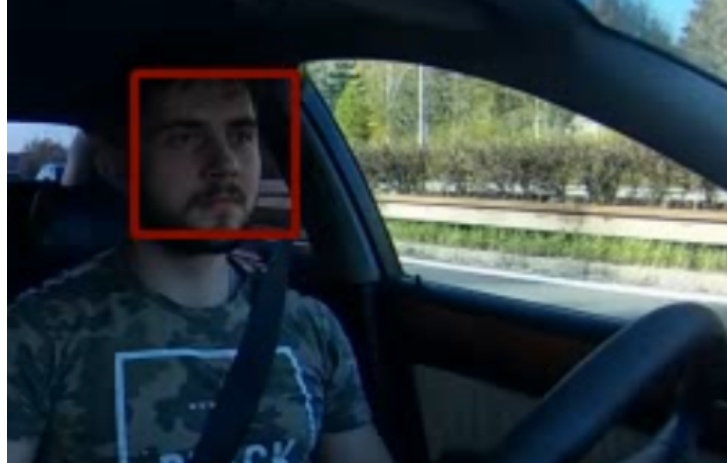
- **Dvoj-obdĺžnikové** (*angl. two-rectangle*) - porovnávajú sumu pixlov v obdĺžnikových oblastiach, ktoré sa nachádzajú vedľa seba, vodorovne, alebo zvislo.
- **Troj-obdĺžnikové** (*angl. three-rectangle*) - porovnávajú sumu obdĺžnikových oblastí, ktoré sa nachádzajú po oboch stranách aktuálnej oblasti a sumu aktuálnej oblasti.
- **Štvor-obdĺžnikové** (*angl. four-rectangle*) - počítajú rozdiel medzi dvoma aktuálnymi obdĺžnikovými областami, ktoré sa dotýkajú svojimi rohmi, a obdĺžnikovými областami medzi nimi.



Obr. 2: Haar - dvoj-obdĺžnikové príznaky (A, B), troj-obdĺžnikové príznaky (C) a štvor-obdĺžnikové príznaky (D) [1]

Znázornenie jednotlivých typov príznakov môžeme vidieť na obrázku 2. Jednotlivé príznaky môžu byť použité dostatočne efektívne. Efektivita klesá pri použití príznaku na celý obraz. Vhodným riešením je preto skombinovať viacero príznakov. Na výber správnych efektívnych príznakov sa používajú špeciálne algoritmy. Jedným z najpoužívanejších algoritmov pre zvýšenie efektivity výberu príznakov je AdaBoost [9], ktorý vytvoril profesor Yoav Freund. Jedná sa o

klasifikačný algoritmus, ktorý je schopný vytvoriť dostatočne silný klasifikátor z kombinácii viacerých slabších klasifikátorov. Metódou Haar je možné detekovať rôzne triedy objektov. Jedným z najčastejších a jednoducho detekovatelných typov objektov je napríklad tvár.



Obr. 3: Haar - detekcia tváre v slabých svetelných podmienkach

## 2.2 HOG

S nápadom vylepšiť detekciu objektov použitím príznakov prišli v roku 2005 Navneed Dalal a Bill Triggs [2], kde postupne vyskúšali niekoľko typov deskriptorov. V práci taktiež podrobne rozobrali možnosti a spôsoby ako správne určiť parametre ich detekčnej metódy pre správne fungovanie detekcie jednotlivých tried. Hlavnou myšlienkou ich metódy je, že objekt môže byť charakterizovaný viacerými spôsobmi. Táto metóda je rozdelená do niekoľkých samostatných krokov (Obr. 4):

- **Úprava obrazu** - v tomto kroku je potrebné v obraze upraviť kontrast a jas, ktoré by mohli spôsobovať problémy v nasledujúcich krokoch. Okrem tejto úpravy je možné obraz upraviť napríklad gamma filtrom.
- **Výpočet gradientov** - veľkosť gradientov sa počíta na základe vstupného obrazu a masky. Masky, ktoré sa používajú v tomto kroku sú  $[-1, 0, 1]$  alebo  $[-1, 0, 1]^T$ . Gradienty je nutné vypočítať v obidvoch osách, čím získame  $I_x$  a  $I_y$ . Po získaní gradientov je potrebné vypočítať veľkosť gradientov  $m(x,y)$  a ich smer  $\theta(x,y)$ :

$$m(x,y) = \sqrt{I_x^2 + I_y^2} \quad (2)$$

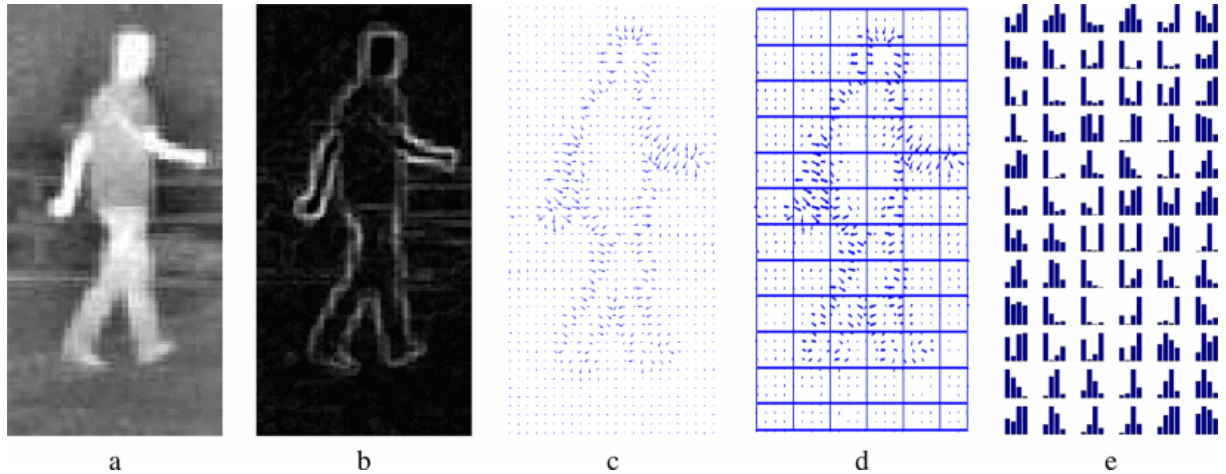
$$\theta(x,y) = \left( \frac{I_y}{I_x} \right) \quad (3)$$

- **Normalizácia** - Pre správne fungovanie je potrebné obraz normalizovať, aby sa minimalizovali rozdiely medzi jednotlivými bunkami. Tento krok spočíva v skladaní viacerých buniek, čím následne vznikajú bloky.
- **Deskriptor** - Je vytvorený zo vstupného obrazu do jednotlivých blokov. Jednotlivé bloky sa posúvajú a prekrývajú o daný počet pixelov. Výsledok deskriptoru je odovzdaný klasifikátoru, ktorý následne určuje do akej triedy objekt patrí. Jeden z často používaných klasifikátorov je Support vector machine (SVM), ktorý napríklad používali autori práce na efektívnu detekciu chodcov. [10]



Obr. 4: HOG - séria krokov [2]

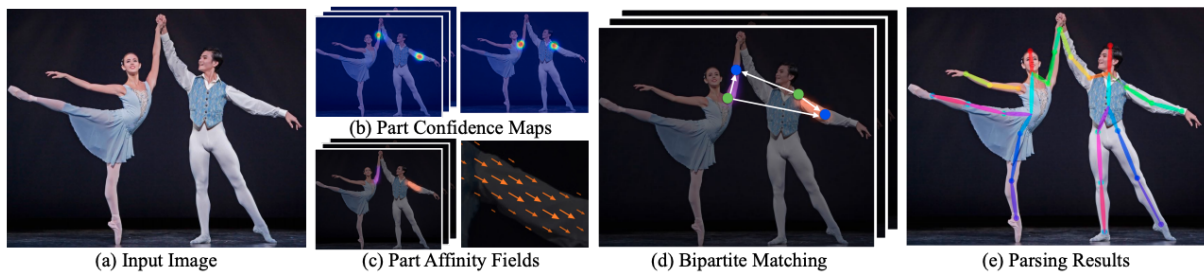
Výstupom tejto metódy je pole histogramov pre jednotlivé bloky. Histogram predstavuje grafické rozloženie intenzity jasu vstupného obrazu. Pri niektorých špecifických obrazoch je potrebné histogram vyrovnať. Tento krok je potrebný najmä pri obrazoch, ktoré su príliš tmavé alebo príliš svetlé. Pomocou vyrovnenia (*angl. equalization*) histogramu dokážeme zvýšiť kontrast obrazu. Jednotlivé kroky znázornené na vstupnom obraze chodca môžeme vidieť na obrázku 5



Obr. 5: HOG - vstupný obraz (a), normalizácia gradientu (b), orientácia gradientu (c), rozdelenie do buniek (d) vypočítané histogramy (e). [3]

## 2.3 OpenPose

OpenPose [4] je framework, ktorý bol prvýkrát uvedený verejnosti už v roku 2016. Detekcia ľudského postoja predstavuje hlavný problém s lokalizáciou častí ľudského tela, ako sú ramená, lakty a členky zo vstupného obrázka alebo videa. Vo väčšine dnešných aplikácií detekcie postačujú v reálnom svete sa vyžaduje vysoký stupeň presnosti, ako aj spracovanie v reálnom čase. OpenPose, ktorý bol vyvinutý výskumníkmi na univerzite Carnegie Mellon University, možno považovať za najmodernejší prístup pri detekcii ľudských v reálnom čase. Jedná sa o open-source projekt, ktorého zdrojové kódy sú dostupné na oficiálnej Github stránke [11].



Obr. 6: OpenPose - odhad viacerých ôsob v reálnom čase pomocou polí afinity filtra. [4]

Samotný framework je veľmi detailne vysvetlený a dobre zdokumentovaný. OpenPose bol pôvodne napísaný v C++ a Caffe [12]. Postupom času však autori vytvorili aj nadstavbu pre jazyk Python, s ktorým sa rozšírili možnosti jeho využitia medzi ostratnými programátormi. Základná myšlienka detekcie pomocou OpenPose sa skladá z viacerých krokov:

- **Spracovanie vstupného obrazu** - vstupný obrázok (Obr. 6a) privádza ako vstup do „dvojvetvového viacstupňového“ CNN. Dve vetvy znamenajú, že CNN produkuje dva rôzne výstupy z jedného vstupného obrazu. Viacstupňové jednoducho znamená, že sieť je v každej fáze naskladaná jedna na druhú. (Tento krok je analogický jednoduchému zväčšeniu hĺbky neurónovej siete s cieľom zachytiť podstatnejšie výstupy smerom k posledným stupňom.)
- **Spracovanie v dvoch vetvách** - prvá vetva predpovedá mapy dôveryhodnosti (Obr. 6b) rôznych častí tela, ako je pravé oko, ľavé oko, pravé lakty a podobne. Druhá vetva zobrazená modrou farbou predpovedá afinitné polia (Obr. 6c), čo predstavuje stupeň asociácie medzi rôznymi časťami tela.
- **Viacfázové spracovanie** - v prvej fáze sieť vytvorí počiatočnú sadu detekčných máp spoľahlivosti  $S$  a množinu polí afinity častí  $L$ . Potom v každej nasledujúcej fáze predpovede z obidvoch vetiev v predchádzajúcej fáze, spolu s pôvodnými obrazovými znakmi  $F$ , sú zretazené a použité na vytvorenie podrobnejších predpovedí. Pri implementácii OpenPose sa posledná fáza  $t$  zvolí ako číslo 6.

### 2.3.1 Mapa spoľahlivosti

1. vetva v neurónová sieti OpenPose vytvára sadu máp spoľahlivosti  $S$  (rovnica 4). V podstate sa jedná o tabuľku, v ktorej je každej časti tela z datasetu priradená miera spoľahlivosti v rozsahu 0 až 1.

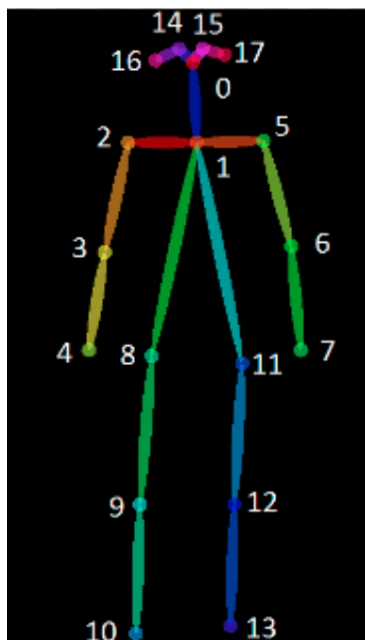
$$S = (S_1, S_2, S_3 \dots S_j) \quad (4)$$

$$S \in \mathbb{R}^{w \times h},$$

$j \in \{1, J\}$ , kde  $J$  je počet všetkých častí tela

Počet častí tela závisí od množiny datasetov, s ktorými je program OpenPose trénovaný. Pokiaľ ide napríklad o súbor datasetu COCO [13],  $J = 19$ , pretože existuje 18 rôznych kľúčových bodov tela + 1 pozadie. Obrázok 7 zobrazuje rôzne časti tela s prideleným ID pre súbor údajov COCO. Pre model trénovaný s dátovým súborom COCO bude sada  $S$  obsahovať prvky  $S_1, S_2, S_3, \dots, S_{19}$ . V tomto príklade predpokladáme, že prvok  $S_1$  zodpovedá mape spoľahlivosti pre kľúčový bod s číslom 0 ktorý zodpovedá nosu (Obr. 7).

Pre ľahšiu predstavu predpokladáme, že celý obraz má šírku a výšku 5px, čo vedie k vytvoreniu mapy spoľahlivosti o veľkosti 5 X 5. Vo vstupnom obrázku sa nachádza iba jedna tvár. Preto pre mapu spoľahlivosti  $S_1$  (zodpovedajúca za detekciu nosa) vidíme hodnoty s vysokou spoľahlivosťou iba v oblasti, kde sa nos nachádza.



Obr. 7: COCO - označenie častí tela v COCO datasete. [5]



### 2.3.2 Part affinity fields (PAF)

Druhá vetva neurónovej siete vytvára množinu čiastkových afinitných polí  $L$  (rovnica 5).

$$L = (L_1, L_2, L_3 \dots L_c) \quad (5)$$

$$L \in \mathbb{R}^{w \times h \times 2},$$

$c \in \{1, C\}$ , kde  $C$  je počet všetkých končatín

Celkový počet končatín a párov, závisí od datasetu, s ktorým je OpenPose trénovaný. Kvôli prehľadnosti sa uvádzajú dvojice častí tela ako končatiny, napriek tomu, že niektoré páry častí tela nie sú v skutočnosti končatinami (Napríklad oko-nos, ucho-oko atď). Pre dataset COCO je počet párov končatín,  $C = 19$ . Môžeme si predstaviť, že každý prvok v množine  $L$  je mapa veľkosti  $w \times h$ , kde každá bunka obsahuje 2D vektor predstavujúci smer párových prvkov. Napríklad na obrázku 7 môžeme vidieť, že pár častí tela pozostáva z pravého ramena k pravému laktu. Schéma potom ukazuje smerový vektor, ktorý ukazuje z pravého ramena na pravý lakeť. Celý zoznam párov končatín môžeme vidieť vo výpise 1.

---

**COCO\_PAIRS** = [(1, 2), (1, 5), (2, 3), (3, 4), (5, 6), (6, 7), (1, 8), (8, 9),  
(9, 10), (1, 11), (11, 12), (12, 13), (1, 0), (0, 14), (14, 16), (0, 15),  
(15, 17), (2, 16), (5, 17)]

---

Výpis 1: Množina párov končatín v datasete COCO

Okrem Datasetu COCO Dokáže OpenPose pracovať aj s mnohými ďalšími datasetmi. OpenPose bol skúšaný a trénovaný napríklad s datasetmi MPI, BODY\_25 alebo BODY\_25b. Datasety sa líšia vo veľkosti, rýchlosti, ale napríklad aj v presnosti samotnej detekcie. Jednotlivé datasety majú medzi sebou nasledujúce rozdiely:

- **COCO** - je starší dataset, na ktorom bol OpenPose pôvodne vyvíjaný. Postupne sa však nahradzuje novými a modernejšími datasetmi. Jeho výhodou je, že vyžaduje menej pamäte na GPU (schopnosť pracovať s 2 GB GPU a predvoleným nastavením) a pri režime CPU pracuje rýchlejšie oproti novšiemu BODY\_25.
- **BODY\_25** - jedná sa o novší dataset, ktorý je rýchlejší, presnejší a obsahuje ďalšie trénovacie dáta k častiam tela, ktoré nie sú obsiahnuté v COCO datasete (Např. chodidlá). Jeho nevýhodou sú hlavne vysoké hardwarové nároky.
- **MPI** - je určený pre ľudí, ktorí požadujú štruktúru datasetu MPI. Je tiež pomalší oproti BODY\_25 a oveľa menej presný

## 2.4 TensorFlow

tensorflow

## 2.5 Ostatné metódy

WrnchAI je pomerne nový framework, nie je opensource

### 3 Detekcia vodiča vo vozidle

Detekcia objektov v uzavretom priestore (v našom prípade vo vozidle) sa od bežnej detekcie v základných princípoch nelíši. Obmedzenia nastávajú najmä nevhodnou pozíciou alebo natočením kamery. Ak je kamera nesprávne umiestnená, alebo má nedostatočný uhol záberu, nemusí byť zdetekovaný celý snímaný objekt, čo môže viesť k chybe pri jeho detekcii. Autori v práci [14], kde sa zameriavali na oblasť tváre vodiča použili kameru umiestnenú v oblasti stredu palubnej dosky. Tým dosiahli takmer priame natočenie kamery na vodiča, bez toho, aby ho táto kamera výraznejšie obmedzovala vo výhlade. Keďže táto práca bola zameraná hlavne na detekciu očí a úst, nebolo potrebné používať kameru s vysokým uhlom záberu. Mnoho podobných riešení zameraných na snímanie aktivity vodiča sa snaží zaviesť mnoho výrobcov automobilov. Príkladom môže byť výrobca automobilov značky BMW, ktorý v príplatkovej výbave ponúka kameru na snímanie správania vodiča. Táto kamera dokáže reagovať napríklad na únavu alebo zatvorenie očí vodiča. Tento systém dokáže dokonca reagovať aj na to, že vodič má otočenú hlavu a nesleduje premávku. V každom takomto prípade je vodič upozornený zvukovým znamením. Kamera je umiestnená na mieste, odkiaľ je na tvár vodiča priamy výhľad (Obr. 8). Okrem jej vhodného umiestnenia je navyše vybavená aj IR LED diódami, vďaka ktorým kamera funguje aj za znížených svetelných podmienok a v noci, kedy sa vyskytuje najväčší počet mikrosprávok u vodičov vozidiel.



Obr. 8: BMW driving assistant - umiestnenie kamery na snímanie vodiča. [6]

Táto diplomová práca je však zameraná aj na detekciu celého tela vodiča a nielen jeho tváre.

## 4 Využitie sférických kamier na detekciu obrazu

## 4.1 Použitie v analýze videa

- problem s rozlíšením , skreslením , formatom atd.. -

## 4.2 Technické parametre

Go PRO:

THETA:

Senzor FishEye CMOS 2x12MPix Maximálne rozlíšenie (Video) 4K 30fps Maximálne rozlíšenie (Fotografia) 14.5MP (5376x2688px) Svetelnosť f2.0 Vnútorná pamäť 19GB

## 5 Program

- Popis programu
- jazyk Python
- pouzite kniznice

## 5.1 Požiadavky a návrh programu

- požiadavky
- Architektura programu , schemy



## 5.2 Detekcia vodiča

- umiestnenie kamery, detekcia vodiča
- vyber TF vs OpenPose

### 5.2.1 Neurónová sieť

- NN klasifikator
- trenovanie
- testovanie

### 5.3 Orientácia hlavy

- Haar priznaky ,
- prevod 2D na 3D
- smerova priamka

## 5.4 Výstup programu

obrazky

tabulky

## 5.5 Porovnanie výsledkov

porovnanie

## 5.6 Využitie zozbieraných dát

použitie v budúcnosti

## 5.7 Používatelská příručka

`python program.py --use-openPose=true`

## 6 Možnosti vylepšenia detekcie

Zhrnutie výsledkov



## **7 Záver**

Zhrnutie výsledkov

## Literatúra

- [1] Paul Viola, Michael Jones, et al. Robust real-time object detection. *International journal of computer vision*, 4(34-47):4, 2001.
- [2] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.
- [3] Massimo Bertozzi, Alberto Broggi, Mike Del Rose, Mirko Felisa, Alain Rakotomamonjy, and Frédéric Suard. A pedestrian detector using histograms of oriented gradients and a support vector machine classifier. In *2007 IEEE Intelligent Transportation Systems Conference*, pages 143–148. IEEE, 2007.
- [4] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.
- [5] Bryan Chung. Openpose in processing and opencv (dnn), 2018. [Online; Citované 23.02.2020 z <http://www.magicandlove.com/blog/2018/08/06/openpose-in-processing-and-opencv-dnn/>.
- [6] Gabriel Nica. Bmw assisted driving view looks cool and useful, 2019. [Online; Citované 16.02.2020 z <https://cdn.bmwblog.com/wp-content/uploads/2019/09/bmw-assisted-driving-view.jpg>.
- [7] Azriel Rosenfeld. Picture processing by computer. *ACM Computing Surveys*, 1(3):147–176, Jan 1969.
- [8] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.
- [9] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.
- [10] Yanwei Pang, Yuan Yuan, Xuelong Li, and Jing Pan. Efficient hog human detection. *Signal Processing*, 91(4):773–781, 2011.
- [11] CMU-Perceptual-Computing-Lab. Openpose: Real-time multi-person keypoint detection library for body, face, hands, and foot estimation, 2017. [Online; Citované 23.02.2020 z <https://github.com/CMU-Perceptual-Computing-Lab/openpose>.

- [12] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678, 2014.
- [13] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [14] Paul Smith, Mubarak Shah, and Niels da Vitoria Lobo. Determining driver visual attention with one camera. *IEEE transactions on intelligent transportation systems*, 4(4):205–218, 2003.