

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Analýza řidiče za pomocí sférických kamer

Driver Analysis Using Spherical Cameras

2020

Michal Falát

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 podakoval 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 jednotlivych knižníc a metód, ktoré sa používajú na analýzu ľudského tela a tváre v obraze. Posledná časť je venovaná porovaniu 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 datas captured by spherical camera and summary of results.

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

Obsah

Zoznam použitých skratiek a symbolov	7
Zoznam obrázkov	8
Zoznam výpisov zdrojového kódu	9
1 Úvod	10
2 Detekcia a analýza ľudského tela v obrazoch	11
2.1 Haar	11
2.2 HOG	13
2.3 OpenPose	15
2.4 TF Pose Estimation	18
2.5 AlphaPose	20
2.6 Ostatné metódy	21
3 Detekcia vodiča vo vozidle	23
4 Využitie sférických kamier na detekciu obrazu	24
4.1 Použitie v analýze videa	25
4.2 Technické parametre	25
5 Program	26
5.1 Požiadavky a návrh programu	27
5.2 Detekcia vodiča	28
5.3 Orientácia hlavy	30
5.4 Výstup programu	31
5.5 Porovnanie výsledkov	32
5.6 Využitie zozbieraných dát	33
5.7 Používateľská príručka	34
6 Možnosti vylepšenia detekcie	35
7 Záver	36
Literatúra	37

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
NMS	– Non maximum suppression
OpenCV	– Open source computer vision
PAF	– Part affinity fields
PX	– Pixel
RCNN	– Region convolutional neural network
SPPE	– Single-person pose estimator
SSTN	– Symmetric spatial transformer network
TF	– Tensorflow
VR	– Virtual reality

Zoznam obrázkov

1	Výpočet integrálneho obrazu - vstupný obraz (A) , integrálny obraz (B)	11
2	Haar - dvoj-obdlžnikové príznaky (A, B), troj-obdlžnikové príznaky (C) a štvorobdlžnikové 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 affinity filtra. [4] .	15
7	COCO - označenie častí tela v COCO datasete. [5]	16
8	Tepelná mapa - vstupný obraz (a), tepelná mapa postavy (b)	19
9	Problém redundantnej detektie s použitím metódy bounding box [6]	20
10	WrnchAI - vstupný obraz (a), výsledok spracovania na platforme WrnchAICloud(b)	22
11	BMW driving assistant - umiestnenie kamery na snímanie vodiča. [7]	23

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 klúčová. 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 situáciach. 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 smrtelným 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 umiestnejenej 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 záchytenie 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 vzdialenosťi 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 sférickú kameru. Počas práce som mal k dispozícii viaceré kamery, s ktorými som zhotovil niekoľko desiatok videí v rôznych situáciach. Z takýchto videí som dokázal analyzovať a zistiť mnoho užitočných informácií, ktore sú spracované v tejto diplomovej práci. Tieto informácie som zbieraním nahrávaním videa sférickymi kamerami za rôznych svetelných podmienok a pozícii 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, rozhodo lso msa 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 sférickou 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 [8] rozoberal napríklad jednoduchú analýzu obrazu a spracovanie obrazov s dostupnými prostriedkami. Postupom času sa však počítacová 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 trénovacích dát. Vo svojej práci som pracoval hlavne s metódami Haar a HOG, ktoré sa radia medzi najpoužívanejšie tradičné metódy a sú im venované samostatné podkapitoly 2.1 a 2.2.

Každý obraz sa skladá z pixelov. Analýza obrazu však nespočíva v prehľadávani jednodlivých 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 [9] 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 samostaných časťí: 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 konkrétné súradnice (x, y) spočíva v súčte hodnôt jasu vľavo a nad súradnicami (x, y) . Výpočet je znározený v rovnici 1.

1	1	1
1	1	1
1	1	1

(A)

1	2	3
2	4	6
3	6	9

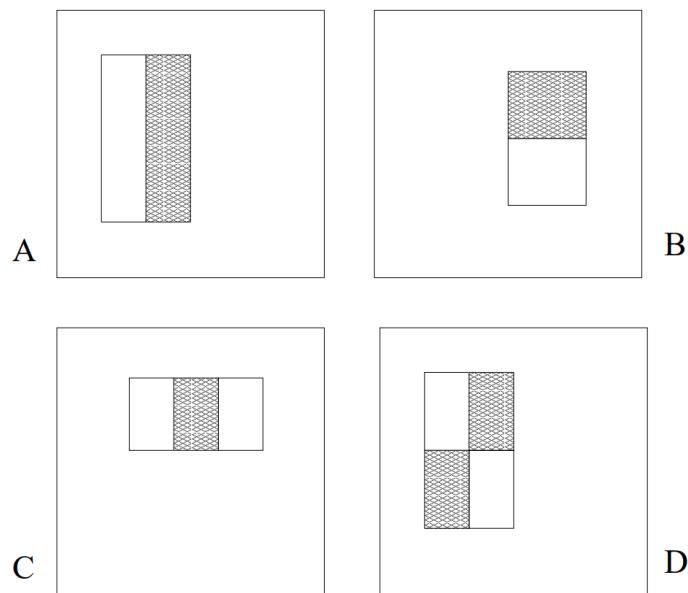
(B)

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 porovnanie celých blokov pixelov. Tieto bloky (častokrát 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:

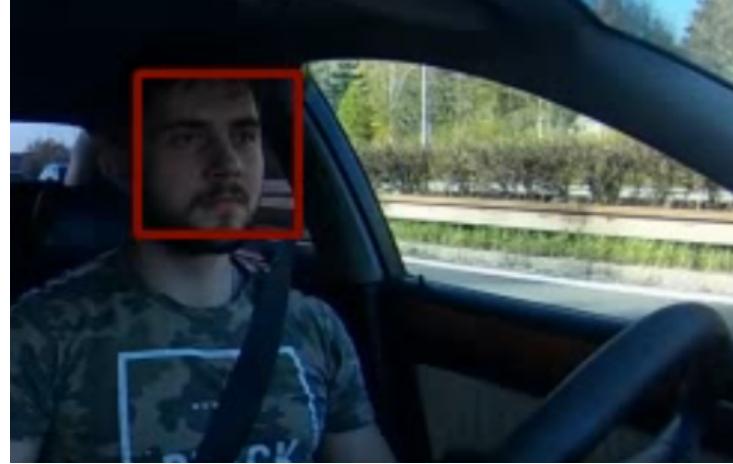
- **Dvoj-obdĺžnikové** (*angl. two-rectangle*) - porovnávajú sumu pixlov v obdĺžnikových oblastiach, ktoré sa nachádzajú vedľa seba, vodorovne, alebo zvislo.
- **Troj-obdĺžnikové** (*angl. three-rectangle*) - porovnávajú sumu obdĺžnikových oblastí, ktoré sa nachádzajú po oboch stranách aktuálnej oblasti a sumu aktuálnej oblasti.
- **Štvor-obdĺžnikové** (*angl. four-rectangle*) - počítajú rozdiel medzi dvoma aktuálnymi obdĺžnikovými oblasťami, ktoré sa dotýkajú svojimi rohmi, a obdĺžnikovými oblasťami 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 [10], ktorý vytvoril profesor Yoav Freund. Jedná

se o klasifikačný algoritmus, ktorý je schopný vytvoriť dostatočne silný klasifikátor z kombinácie viacerých slabších klasifikátorov. Metódou Haar je možné detektovať rôzne triedy objektov. Jedným z najčastejších a jednoduchu 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 Navneet 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 urpavíť napríklad gamma filtrom.
- **Výpočet gradientov** - veľkosť gradientov sa počíta na základe vstupného obrazu a masky. Masky, ktoré sa používaju v tomto kroku sú $\begin{bmatrix} -1, & 0, & 1 \end{bmatrix}$ alebo $\begin{bmatrix} -1, & 0, & 1 \end{bmatrix}^T$. 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 $\theta(x,y)$:

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

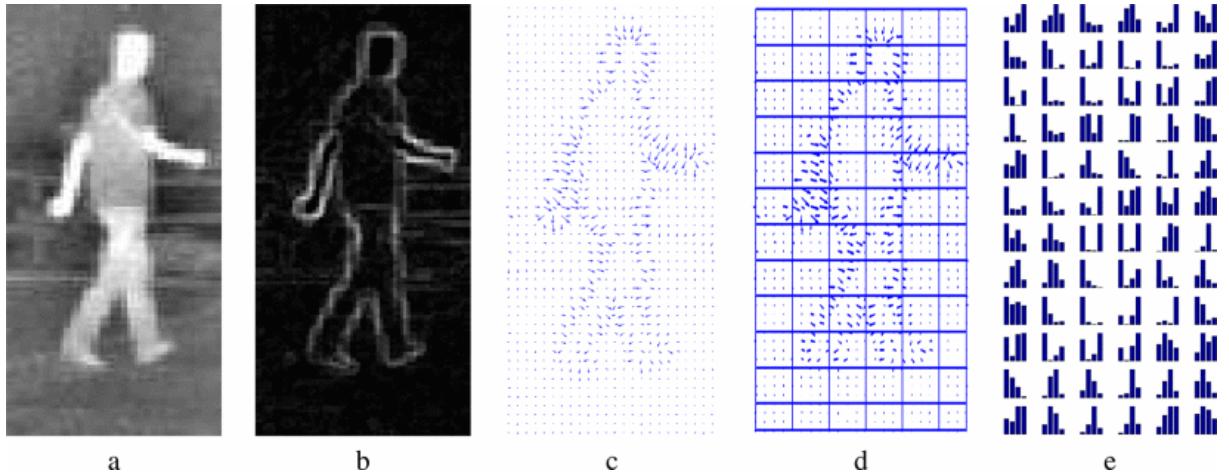
$$\theta(x,y) = \left(\frac{I_y}{I_x} \right) \quad (3)$$

- **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 prekryvajú 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. [11]



Obr. 4: HOG - séria krokov [2]

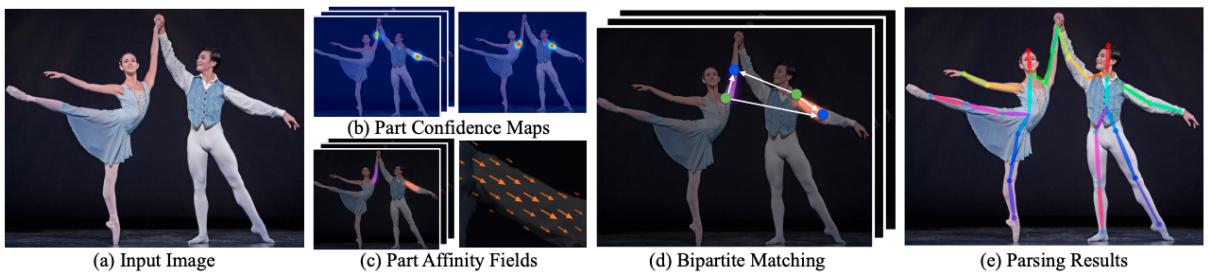
Výstupom tejto metódy je pole histogramov pre jednotlivé bloky. Histogram predstavuje grafické rozloženie intenzity jasu vstupného obrazu. Pri niektorých špecifických obrazoch je potrebné histogram vyrovnať. Tento krok je potrebný najmä pri obrazoch, ktoré sú príliš tmavé alebo príliš svetlé. Pomocou vyrovnania (*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 prvýkrat uvedený verejnosti už v roku 2016. Detekcia ľudského postoja predstavuje hlavný problém s lokalizáciou častí ľudského tela, ako sú ramená, lakte a členky zo vstupného obrázka alebo videa. Vo väčšine dnešných aplikácií detekcie postaáv 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 [12].



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 [13]. 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 siet 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, levé oko, pravé lakte 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 siet vytvorí počiatočnú sadu detekčných máp spoločlivosti 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.

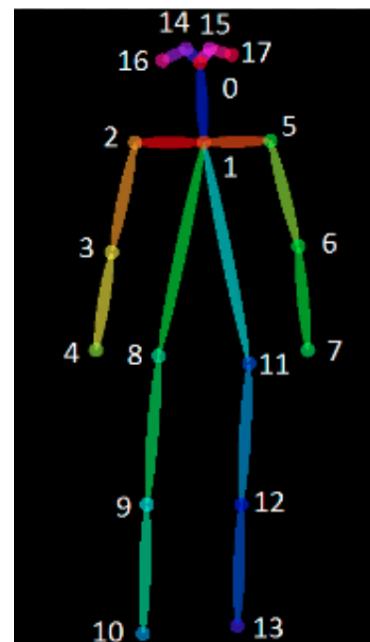
$$S = (S_1, S_2, S_3 \dots S_j) \quad (4)$$

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

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

Počet častí tela závisí od množiny datasetov, s ktorými je program OpenPose trénovaný. Pokiaľ ide napríklad o súbor datasetu COCO [14], $J = 19$, 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 5×5 . Vo vstupnom obrázku sa nachádza iba jedna tvář. 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).

$$\begin{aligned} L &= (L_1, L_2, L_3 \dots L_c) \\ L &\in \mathbb{R}^{w \times h \times 2}, \\ c &\in \{1, C\}, \text{kde } C \text{ je počet všetkých končatín} \end{aligned} \tag{5}$$

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 párs častí tela pozostáva z pravého ramena k pravému laktu. Schéma potom ukazuje smerový vektor, ktorý ukazuje z pravého ramena na pravý lakte. 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. Datasetsy sa líšia vo veľkosti, rýchlosťi, ale napríklad aj v presnosti samotnej detekcie. Jednotlivé datasetsy 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 dátá k časťiam tela, ktoré nie sú obsiahnuté v COCO datasete (Napr. 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 TF Pose Estimation

Tento framework pre detekciu postáv bol implementovaný pomocou knižnice Tensorflow. Po-skytuje tiež niekoľko variantov, ktoré sa odlišujú najmä v zmenách pre spracovanie v reálnom čase na CPU alebo zariadení s nízkou spotrebou. Z tohto dôvodu je možné používať napríklad na mobilných zariadeniach alebo internetových prehladiačoch použitím knižnice tensorflow.js. Tensrflow Pose estimation používa na detekciu vlastný model s názvom PoseNet. PoseNet sa dá použiť na odhad jednej pozície alebo viacerých pozícii v obraze súčasne. To znamená, že existuje verzia algoritmu, ktorý dokáže detektovať iba jednu osobu v obraze a druhá verzia, ktorá dokáže zistiť viac osôb v obraze. Hlavnou výhodou použitím detekcie jednej osoby je rýchlejšie spracovanie a nižší výpočtový výkon. Podstatnou nevýhodou však je, že vyžaduje iba jeden objekt prítomný na obrázku. Pri súčasnej pozícii viacerých osôb v obraze tento algoritmus nedokáže zdetektovať správne ani jednu osobu. Je preto potrebné sa zamyslieť sa hneď na začiatku, koľko ôsob sa reálne môže v obraze nachádzať. Hlavná myšlienka tohto algoritmu sa skladá z dvoch krokov podobne ako pri knižnici OpenPose.

- **Detekcia pozície** - na najvyšej úrovni modelu PoseNet sa vráti objekt, ktorá obsahuje zoznam klúčových bodov a skóre spoľahlivosti pre každú detektovanú osobu.
- **Výpočet spoľahlivosti pozície** - skóre spoľahlivosti určuje celkovú dôveru v odhadovaní pozície. Je v rozsahu od 0 až 1. Môže sa použiť na skrytie pozícii, ktoré sa nepovažujú za dostatočne výrazné.
- **Výpočet klúčových bodov** - odhadované časti tela osoby, ako napríklad nos, pravé ucho, ľavé koleno, pravá noha atď. Obsahuje pozíciu a spoľahlivosť klúčového bodu. PoseNet štandardne zistuje 17 klúčových bodov.
- **Poloha klúčového bodu** - pozostáva z 2D súradníc v pôvodnom vstupnom obrázku, kde bol zistený klúčový bod.

Model PoseNet je nezávislý na veľkosti vstupného obrazu. To znamená, že môže predikovať polohy pozícii v rovnakom rozlíšení ako pôvodný obrázok bez ohľadu na to, či je obraz zmenšený. PoseNet môže byť nakonfigurovaný tak, aby mal vyššiu presnosť na úkor rýchlosťi detekcie na-stavením výstupného kroku. Výstupný krok určuje, do akej miery zmenšujeme výstup vzhľadom na veľkosť vstupného obrázka. To ovplyvňuje veľkosť jednotlivých vrstiev a výstupy modelu. Čím vyšší je výstupný krok, tým menší je počet vrstiev v sieti a výstupoch a tým aj ich presnosť. V základnej implementácii môže mať výstupný krok hodnoty 8, 16 alebo 32. Inými slovami, výstupný krok 32 bude mať za následok najrýchlejší výkon, ale najnižšiu presnosť, zatiaľ čo 8 bude mať najvyššiu presnosť, ale najpomalší čas detekcie.

2.4.1 Výstup

Ked' PoseNet spracováva obraz, v skutočnosti vytvára tepelnú mapu (*angl. Heatmap*) spolu s ofsetovými vektormi, ktoré je možné dekódovať, aby sa v obraze našli oblasti s vysokou spoľahlivosťou. Veľkou výhodou pri detekcii postáv v obraze cez Tensorflow je teda to, že spolu s vektormi výstupného modelu dostávame aj štrukturované dátá pravdepodobnosti jednotlivých častí postavy, ktoré môžeme využiť na ďalšie spracovanie, alebo znázorniť (Obr. 8).

2.4.2 Tepelná mapa

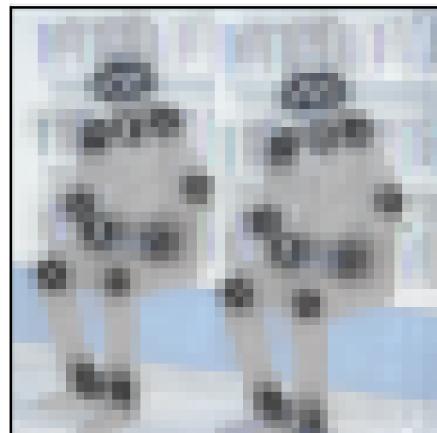
Každá pozícia v tejto tepelnej mape má určité skóre spoľahlivosti. Tento údaj vyjadruje pravdepodobnosť, že v danom umiestnení existuje určitá časť daného typu. Dá sa to považovať za rozdelenie pôvodného obrázka do mriežky 15×15 , kde skóre v termografickej mape poskytuje klasifikáciu pravdepodobnosti, že každý kľúčový bod existuje v každom štvorci mriežky.

2.4.3 Výstupný vektor

Každý ofsetový vektor je 3D vektor, ktorý má veľkosť Šírkaobrazu \times výškaobrazu \times 34. Číslo 34 je dvojnásobok kľúčových bodov ($2 * 17$). Tepelné mapy sú iba aproximáciou toho, kde sa skutočne kľúčové body nachádzajú, pričom výstupné vektory zodpovedajú svojou polohou bodom tepelnej mapy a používajú sa na predpovedanie presnej polohy jednotlivých častí ľudského tela. Prvých 17 rezov výstupného vektora obsahuje x-ovú súradnicu vektora a posledných 17 rezov y-ovú súradnicu. Veľkosti výstupného vektora sú v rovnakej mierke ako pôvodný vstupný obrázok.



(a)



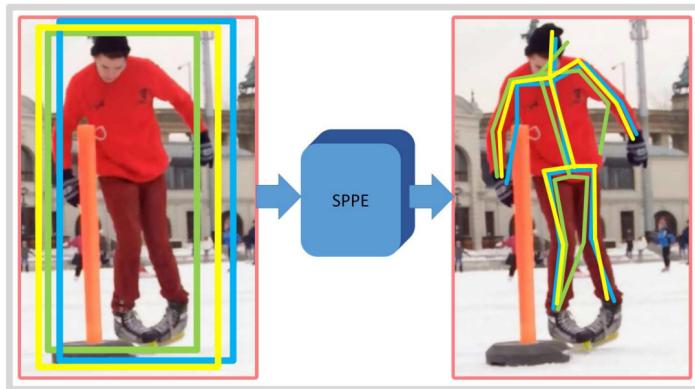
(b)

Obr. 8: Tepelná mapa - vstupný obraz (a), tepelná mapa postavy (b)

2.5 AlphaPose

AlphaPose [6] je framework, ktorý vznikol v roku 2017. Je zameraný na detekciu ľudských postáv v čo najpresnejšej miere. Tento framework pracuje na princípe používania ohraničovacích boxov (*angl. bounding box*). Pre efektívne fungovanie algoritmu je použitý najmodernejší detektor objektov. Autori práce použili natrénovaný model Faster RCNN [15] a model pre detekciu postáv v obraze [16], ktorý je zameraný na SPPE detekciu. Primárna myšlienka práce spočíva v riešení dvoch hlavných problémov, ktoré pri takejto detekcii vznikajú:

- **Problém lokalizačnej chyby** - skutočnosti je SPPE dosť náchylný na chyby pri spočívajúce v použití ohraničovacích boxov. Aj v prípadoch, keď sú ohraničovacie rámcinky sú považované za správne a majú dostatočne vysokú spoľahlivosť ($I_oU > 0,5$), zistené ľudské pozície môžu byť stále nesprávne.
- **Redundantná detekcia** - SPPE vytvára pózu pre každý ohraničovací box, čo vo výsledku vedie k duplicitnej detekcii rovnakej osoby (Obr.9)



Obr. 9: Problém redundantnej detekcie s použitím metódy bounding box [6]

Na vyriešenie uvedených problémov je použitý model na regionálnu detekciu viacerých postáv (RMPE). Vďaka tomu framework vylepšuje výkon algoritmov na odhadovanie ľudských postojov založených na SPPE. Autori práce tiež navrhli novú symetrickú sieť pre priestorovú detekciu (SSTN), ktorá je pripojená k SPPE na extrakciu jednotlivej osoby z oblasti nepresného ohraničovacieho boxu. Na optimalizáciu tejto siete je zavedená ďalšia paralelná vetva SPPE. Na riešenie problému redundantnej detekcie, je použitý parametrický NMS, ktorý eliminuje nadbytočné pózy pomocou novej metriky vzdialenosť a porovnanie podobnosti pózy. Prístup založený na údajoch je aplikovaný na optimalizáciu parametrov metriky vzdialenosť. RMPE je navrhnutý veľmi všeobecne a práve vďaka tomu je použiteľný pre rôzne ľudské detektory a ďalšie knižnice, ktoré pracujú na princípe single-person detection. AlphaPose používa dataset MPII, s ktorým prekonáva najmodernejšie metódy ako OpenPose. Tento model a zdrojové kódy [17] sú verejne dostupné a určené primárne pre vedu a výskum.

2.6 Ostatné metódy

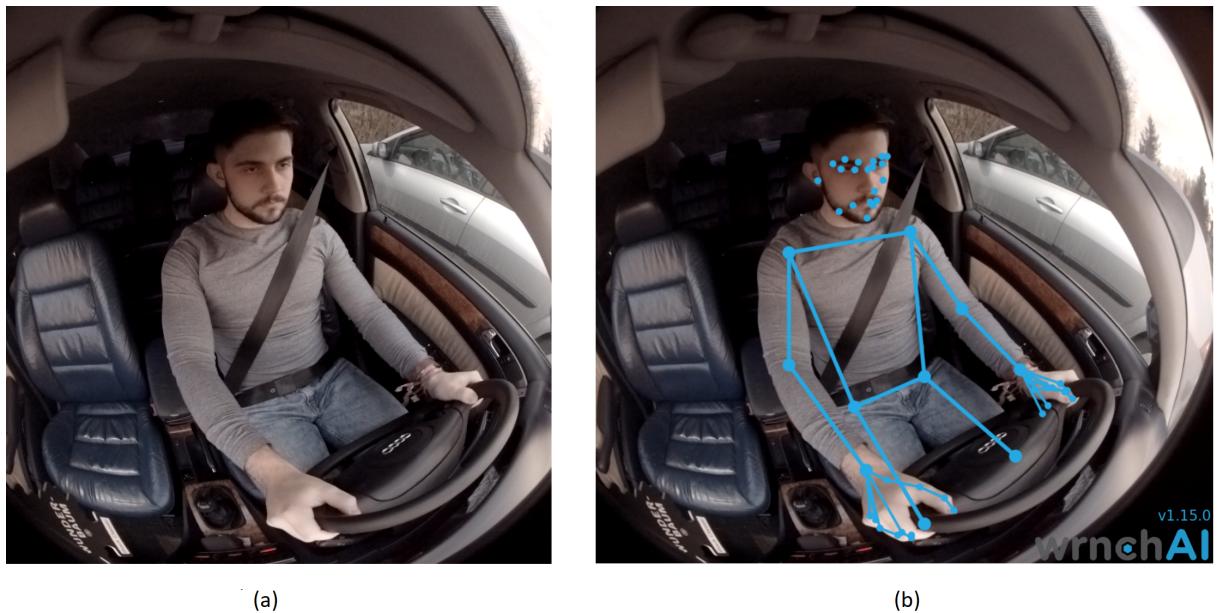
Okrem najpoužívajenších metód ako OpenPose a TF pose estimation, či AlphaPose existuje aj mnoho ďalších riešení. Tieto riešenia často vychádzajú zo základného princípu detektie postáv, ktoré používajú aj tieto hlavné knižnice. Hlavný rozdiel je však použitie upravenej neurónovej siete alebo zmena rôznych parametrov. Vďaka takejto úprave je možné dosiahnuť v špecifických prípadoch vyššiu úspešnosť a presnosť detektie alebo dokonca znížiť výpočtový čas potrebný na detekciu a spracovanie. Tieto riešenia sú však často obmedzované najmä chýbajúcou dokumentáciou, nedostatočnou implementáciou vo viacerých jazykoch, alebo nemožnosťou jednoduchej zmeny konfigurácie. Okrem voľne dostupných riešení existuje aj mnoho projektov, ktoré nie sú OpenSource, čo môže mnoho používateľov zo začiatku odradiť. V nasledujúcom odseku sú zhrnuté a v krátkosti rozpisane ďalšie knižnice, ktoré sa používajú na detekciu postáv v obrazoch.

2.6.1 WrnchAI

WrnchAI je framework, ktorý bol vytvorený už v roku 2014. S myšlienkovou využitím umelú inteligenciou a založiť projekt pre detekciu postáv prišiel Paul Kruszewski. Napriek tomu, že tento framework existuje už niekoľko rokov a v niektorých článkoch [18] dokázal dokonca získať lepsí čas detektie oproti iným metódam, nestal sa natoľko populárny medzi vývojarmi. Jedným z hlavných dôvodov je to, že knižnica nie je voľne dostupná pre vývojárov a jedná sa o platený projekt bez voľného prístupu k zdrojovému kódu. Súčasná cenová politika základnej verzie sa pohybuje na úrovni \$500 za mesiac používania s možnosťou vyskúšania trial verzie na 1. mesiac zadarmo. Trial verzia však funguje v obmedzenom režime bez prístupu ku všetkým platformám. Pre bežného človeka sa táto cena môže zdať privysoká, avšak vzhľadom na jeho široké možnosti využitia v rôznych oblastiach sa jedná o adekvátnu cenu. WrnchAI však našiel svoje uplatnenie možnosťou využitia na rôznych typoch zariadení, ktoré sa prispôsobujú konkrétnym podmienkam. V súčasnosti WrnchAI poskytuje 4 hlavné platformy:

- **wrnchPC** - umožňuje zachytiť ľudský pohyb od toľkých ľudí, kolko chcete, bez toho, aby ste museli nosiť tažkopádne senzory. Náš tretí AI motor je agnostický fotoaparát, hardvérový AI hardvér, vyrobený tak, aby sa ľahko integroval do všetkých aplikácií. Používame spoľahlivé modely hlbokého učenia, aby sme zabezpečili spoľahlivé zachytenie ľudského pohybu pre všetky činnosti a v akomkoľvek prostredí
- **wrnchCloud** - detekuje ľudský postoj a pohyb z akéhokoľvek video vstupu. WrnchCloud je nákladovo veľmi efektívne riešenie, pretože nepotrebuje žiadne vysoké hardwarové nároky okrem bežného počítača s pripojením na internet. Celý proces detektie prebieha na serveri, čo odbremenuje zákazníka od nákupu drahých grafických kariet a podobne. Toto riešenie je škálovateľné s možnosťou spracovania viacerých vstupov súčasne. Využitie clouдовého riešenia zároveň pomáha držať krok moderných trendov v internetových technológiách.

- **wrnchMobile** - umožňuje využiť mobilné zariadenia na vytvorenie programu snímania postáv v reálnom čase. Taktiež dokáže spolupracovať s mobilnými operačnými systémami ako Android alebo IOS. Jeho základnym princípom je efektívne využitie výpočtového výkonu na malých prenosných zariadeniach ako sú napríklad mobilné telefóny. Tento druh platformy taktiež dokáže pracovať aj s ďalšími pokročilými vecami ako je napríklad rozšírená realita. S touto pomocou rozšírenej reality dokáže wrnchMobile snímanú postavu detektovať a zároveň zobrazovať napríklad do VR okuliarov.
- **wrnchEmbedded** - poskytuje možnosť detektovať postavy na rôznych automatizovaných zariadeniach ako napríklad roboty, samoriadiace vozidlá, aby mohli v reálnom čase vidieť a interagovať s ľuďmi a ich pohybom. WrnchEmbedded pomáha zariadeniam predvídať ľudské správanie a najlepšie reagovať vo všetkých situáciách. Ich program je optimalizovaný najmä na rôzne priemyselné platformy a počítače, vďaka ktorým by mala byť ich integrácia s platformou Wrnch veľmi jednoduchá. V Súčasnosti je však táto časť platformy stále v aktívnom vývoji a nie je dostupná.



Obr. 10: WrnchAI - vstupný obraz (a), výsledok spracovania na platforme WrnchAICloud(b)

Na obrázku 10 môžeme vidieť výsledok metódy cloubovej platformy WrnchAI. Táto platforma funguje dostatočne spoľahlivo aj aj pri atypických polohách ľudského tela, ako je napríklad sedavá poloha vodiča za volantom. WrnchAI okrem polohy tela poskytuje aj detekciu klúčových bodov tváre a klúčové body rúk a prstov. Oproti ostatným knižniciam a frameworkom je možnosť použiť WrnchAI ako komplexne riešenie na detekciu celého ľudského tela, čo napríklad pri OpenPose nieje bez použitia ďalších rozšírené možné. V skúšobnej trial verzii taktiež nieje možné odstrániť vodoznak firmy, ktorý sa nachádzav pravom dolnom rohu výstupného obrázka.

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 zdetektovaný celý snímaný objekt, čo môže viesť k chybe pri jeho detekcii. Autori v práci [19], kde sa zameriaval na oblasť tváre vodiča použili kamery umiestnené v oblasti stredu palubnej dosky. Tým dosiahli takmer priame natočenie kamery na vodiča, bez toho, aby ho táto kamera výraznejšie obmedzovala vo výhľade. Kedže táto práca bola zameraná primárne na detekciu očí a úst, nebolo potrebné používať kameru s vysokým uhlom záberu. Mnoho podobných riešení zameraných na snímanie aktivity vodiča sa snažia zaviesť aj výrobcovia 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. Vo všeobecnosti tento systém dokáže dokonca reagovať aj na to, že vodič má otočenú hlavu a nesleduje premávku, čo môže viesť k nebezpečnej situácii. V každom takomto prípade je vodič upozornený zvukovým znamením. Kamera je umiestnená v prístrjovej doske na mieste, odkiaľ je na tvár vodiča priamy výhľad (Obr. 11). 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 mikrospánkov u vodičov všetkých vozidiel. Vďaka takému riešeniu sa dokáže predísť mnohým nebezpečným situáciám a dopravným nehodám.



Obr. 11: BMW driving assistant - umiestnenie kamery na snímanie vodiča. [7]

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

4 Využitie sférických kamier na detekciu obrazu

Výstup z kamery je reprezentovaný 2D snímkou, ktorá však zachytáva obraz z celého svojho okolia. Medzi najznámejšie projekcie na zachytenie 3D obrazu patrí equirectangular panorama projection, ktoré je definované dvomi uhlami: latitude $\omega \in [-90^\circ, +90^\circ]$ a longitude $\lambda \in [-180^\circ, +180^\circ]$.

4.1 Použitie v analýze videa

- problem s rozlišením , skreslenim , 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 siet

-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

pouzitie v buducnosti

5.7 Používateľská príručka

python program.py –use-openPose=true

6 Možnosti vylepšenia detekcie

Zhrnutie vysledkov

7 Záver

Zhrnutie vysledkov

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] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe - regional multi-person pose estimation. In *ICCV*, 2017.
- [7] 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>].
- [8] Azriel Rosenfeld. Picture processing by computer. *ACM Computing Surveys*, 1(3):147–176, Jan 1969.
- [9] 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.
- [10] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.
- [11] Yanwei Pang, Yuan Yuan, Xuelong Li, and Jing Pan. Efficient hog human detection. *Signal Processing*, 91(4):773–781, 2011.

- [12] 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>.
- [13] 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.
- [14] 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.
- [15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- [16] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.
- [17] AlphaPose. Real-time and accurate multi-person pose estimation and tracking system, 2017. [Online; Citované 26.02.2020 z <https://github.com/MVIG-SJTU/AlphaPose>.
- [18] Vikas Gupta. Pose detection comparison : wrnchai vs openpose, 2019. [Online; Citované 03.03.2020 z <https://www.learnopencv.com/pose-detection-comparison-wrnchai-vs-openpose/>.
- [19] 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.