

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

BAKALÁRSKA PRÁCA

LUKÁŠ VILIM

Počítanie obsadenosti parkoviska pomocou kamery

Vedúci práce: doc. RNDr. Milan Ftáčnik, CSc.

Bratislava 2021

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

BAKALÁRSKA PRÁCA

LUKÁŠ VILIM

Počítanie obsadenosti parkoviska pomocou kamery

Študijný program: Aplikovaná informatika

Študijný odbor: Informatika

Školiace pracovisko: Univerzita Komenského v Bratislave,
Fakulta matematiky, fyziky a informatiky,
Katedra aplikovanej informatiky

Vedúci bakalárskej práce: doc. RNDr. Milan Ftáčnik, CSc.

Bratislava 2021



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Lukáš Vilim
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Počítanie obsadenosti parkoviska pomocou kamery
Counting of occupancy of the parking lot by camera

Anotácia: Na miestach, kde je veľký záujem o parkovanie (ako napr. cez deň pred Matematickým pavilonom FMFI UK), ale nie je tam žiadna regulácia, často motoristi vchádzajú do parkoviska, ktoré je plne obsadené, lebo to pri vjazde nevidno a nemajú žiadny iný signál, že je parkovisko plné.

Cieľ: Cieľom práce je s využitím metód spracovania obrazu navrhnúť a implementovať aplikáciu, ktorá umožní jednoduché počítanie áut prichádzajúcich a odchádzajúcich z parkoviska a porovná ho s vyznačenou kapacitou parkoviska. Úspešnosť aplikácie otestuje na reálnych dátach.

Vedúci: doc. RNDr. Milan Ftáčnik, CSc.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.
Dátum zadania: 06.10.2020

Dátum schválenia: 03.11.2020

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

.....

študent

.....

vedúci práce

Abstrakt

Lukáš Vilim: *Počítanie obsadenosti parkoviska pomocou kamery*. Univerzita Komenského v Bratislave. Fakulta matematiky, fyziky a informatiky. Vedúci: doc. RNDr. Milan Ftáčnik, CSc. Stupeň odbornej klasifikácie: Bakalár.

Abstract

Lukáš Vilim: *Counting of occupancy of the parking lot by camera*. Comenius University in Bratislava. Faculty of mathematics, physics and informatics. Thesis Supervisor: doc. RNDr. Milan Ftáčnik, CSc. Degree of classification: Bachelor.

Obsah

1	Úvod	I
2	Prehľad problematiky	II
2.1	Algoritmus s využitím nízko nákladovej kamery	II
2.1.1	Odčítanie pozadia pomocou GMM (Gaussian Mixture Model)	II
2.1.2	Morfologické operácie	III
2.1.3	Analýza BLOB (Binarily Large Object)	IV
2.1.4	Maďarský algoritmus	V
2.1.5	Kalmanov filter	VI
2.1.6	Implementácia a výsledky počítania	VI
2.2	Algoritmus na počítanie a klasifikáciu vozidiel	VI
2.2.1	Detekcia pozadia	VI
2.2.2	Kalibrácia kamery	VII
2.2.3	Kalibrácia pruhov	VIII
2.2.4	Sledovanie vozidiel	IX
2.2.5	Klasifikácia vozidiel	X
2.2.6	Výsledky počítania	X
2.3	Algoritmus detekcie pohybujúcich sa vozidiel vo videu	XI
2.3.1	Rozlišovanie snímok	XI
2.3.2	Prahovanie	XII
2.3.3	Hľadanie kontúr	XIII
2.3.4	Dilatácia obrazu	XIII
2.3.5	Počítanie	XIV
2.4	Bakalárska práca Detekcia voľných parkovacích miest na ulici v meste [4] ..	XV
3	Literatúra	XVI

1 ÚVOD

V dnešnej dobe sú mestá preplnené autami, čo si veľmi často vyžiada množstvo dopravných nehôd. Často tomu prispievajú jednak nepozornosť vodičov ale taktiež aj nehoda pri hľadaní parkovacieho miesta. Ak by sme však vopred vedeli, že parkovisko, na ktoré chceme ísť parkovať, je plné, vôbec by sme na dané parkovisko nemuseli ani ísť a mohli sa tým vyhnúť zbytočným nehodám. Okrem nehôd pri hľadaní parkovacieho miesta strácame drahocenný čas, o ktorý sme pri hľadaní miesta na plnom parkovisku prišli a v neposlednom rade zvyšujeme spotrebu paliva, čo pri dnešnej klimatickej situácii ničomu dobrému neprispieva. Preto je veľmi rozumné vedieť počet voľných miest na parkovisku. Úloha počítania vozidiel sa vo svete najčastejšie používa pri zisťovaní toku vozidiel na viacprúdovej ceste a prispieva možnosti riadenia dopravy, prípadne k zisťovaniu rýchlosti vozidiel.

2 PREHLAD PROBLEMATIKY

2.1 Algoritmus s využitím nízko nákladovej kamery

Dopravné zápchy sú v dnešnej dobe vo veľkých mestách obrovský problém, ktorý vedie k rastúcim počtom dopravných nehôd. Je preto veľmi veľká potreba riadenia dopravy, aby sa zabránilo týmto preťaženiám, zbytočnému plytvaniu časom a tragickým nehodám. Jedným z riešení je regulácia dopravy optimalizáciou načasovania signálov riadenia dopravy. Výklad tohto nízko nákladového kamerového algoritmu na riadenie dopravného toku na ceste vychádza z publikácie Vehicle Detection, Tracking and Counting z roku 2018 [1]

2.1.1 Odčítanie pozadia pomocou GMM (Gaussian Mixture Model)

Odčítanie pozadia je najjednoduchšia metóda detekcie pohybu na videozáznamoch. V publikácii využili na odčítanie pozadia GMM. GMM je podobný rozdeleniu pravdepodobnosti. Táto metóda v počiatočnom štádiu počíta odchýlku, kovarianciu a priemer každého pixelu na snímke. Po príchode novej snímky sa opäť vypočíta odchýlka, kovariancia a priemer a získa sa kumulatívny priemer. Ak je rozdiel medzi hodnotami aktuálnej snímky a kumulatívneho priemeru väčší ako súčin skutočnej hodnoty a štandardnej odchýlky, potom sa klasifikuje ako popredie.



Obrázok 1 - vstupný videozáznam

GMM pomáha pri efektívnom riešení situácií v pozadí, pretože rýchlosť učenia určuje rýchlosť adaptácie na zmeny osvetlenia v pozadí. Obrázok 2 zobrazuje výsledky odčítania pozadia dosiahnuté pomocou GMM, keď sa použije na vstupný videozáznam na obrázku 1.



Obrázok 2 - výsledok po odčítaní pozadia

2.1.2 Morfologické operácie

V publikácii ďalej použili morfologické operácie. Ako vidno na obrázku 2, po odčítaní pozadia, tam stále existuje nejaký šum, na ktorého odstránenie použili morfologické operácie. Morfológia je v zásade opísaná ako teória a techniky používané na štúdium štruktúry objektu a jeho formy. Pomáha vypočítať geometrické štruktúry objektu. Morfologickému operátorovi sa ako vstup poskytne binárny obraz spolu so štruktúrnym prvkom. Bežne používané morfologické operátory sú dilatácia, erózia, zatváranie a otváranie. Erózia a dilatácia sú dva základné operátory. Štruktúrny prvok pomáha definovať ľubovoľnú susedskú štruktúru. Obrázok 3 zobrazuje konečné výsledky morfológie použitej v tomto algoritme. Používa sa plochý morfologický štruktúrny prvok, ktorý tvorí podstatnú súčasť erózie a dilatácie. Tento štruktúrny prvok pozostáva z binárne hodnoteného susedstva, v ktorom sú do morfologických výpočtov zahrnuté iba pixely so skutočnou hodnotou, zatiaľ čo tie, ktoré majú falošné hodnoty, sú z morfologických výpočtov vyňaté. Stredový pixel, pôvod, slúži ako identifikátor pixelu v spracovávanom obraze.



Obrázok 3 - výsledok po použití morfolgických operácií

2.1.3 Analýza BLOB (Binarily Large Object)

Ďalšou zo základných techník počítačového videnia je BLOB analýza. Pri tejto technike je pozorovaný objekt jasne rozoznatelný od pozadia. Táto technika poskytuje dobrý výkon a lepšiu flexibilitu. BLOB je v podstate spojená oblasť a oblasť je ľubovoľná podmnožina obrazových pixelov. Základný koncept tejto techniky vyžaduje, aby sa najskôr oblasť zodpovedajúca predmetu záujmu získala pomocou technik prahovania obrazu, čiže odčítania pozadia. Po druhé, použitím morfolgických operátorov sa odstráni šum, aby sa upravila oblasť. Nakoniec sa upravená oblasť podrobí matematickým meraniam a výpočtom na získanie konečných výsledkov, ako je na obrázku 4. Tieto merania a výpočty sa dosahujú pomocou kaskádového klasifikátora, ktorý využíva skutočnosť, že pomer šírky a výšky pre vozidlá je vždy väčšie ako 1 a že pre ľudí je tento pomer vždy menší ako 1.



Obrázok 4 - Analýza BLOB

Tieto merania a výpočty teda pomáhajú skôr pri rozlišovaní pohybujúcich sa dopravných prostriedkov ako človeka. Detekcia BLOB sa aplikuje iba na vozidlá vstupujúce do oblasti záujmu, ktorá je definovaná detekčným poľom (obrázok 4).

2.1.4 Maďarský algoritmus

Maďarský algoritmus je v podstate model odhadu nákladov, v ktorom je priradenie založené na základe nákladov. Vo väčšine prípadov to môže byť v závislosti od požiadavky maximálny odhad nákladov alebo minimálny odhad nákladov. Maďarský algoritmus možno najlepšie pochopiť maticovou interpretáciou. Predpokladajme, že chceme priradiť štyri stopy na základe priradenia minimálnych nákladov k štyrom detekciám.

Písanie vo forme matice:

m1	m2	m3	m4
n1	n2	n3	n4
o1	o2	o3	o4
p1	p2	p3	p4

Tu máme štyri stopy m, n, o a p a celkovo štyri detekcie. Úlohou je nájsť najmenšie náklady na priradenie stôp k detekciám. Pretože sú riadky a stĺpce matice rovnaké, môžeme usudzovať, že k jednej detekcii je možné priradiť vždy iba jednu stopu.

V prvom kroku odčítame najmenší prvok celého riadku od prvkov toho istého riadku. Výsledkom je najmenej jedna nula v celom riadku. Postup sa opakuje pre všetky riadky matice.

0	m2	m3	m4
n1	n2	0	n4
o1	o2	o3	0
p1	0	p3	p4

Týmto spôsobom sa teda pridelovanie nákladov uskutočňuje v maďarskom algoritme. Pokiaľ ide o maticu vyššieho rádu, ak v určitom okamihu operácie riadkov a stĺpcov neprinášajú plán priradenia, pokračujeme ďalej v aplikovaní operácií riadkov a stĺpcov, až kým nebudeme schopní rozlišovať medzi stopami z hľadiska najmenších nákladov. Takže použitie tohto algoritmu na priradenie stôp detegovaným vozidlám

pomáha rozlíšiť medzi detegovanými a nezistenými vozidlami na videozázname. Hlavným problémom asociácie údajov je určiť korešpondenciu medzi detekciami a viacpásovými vozidlami.

2.1.5 Kalmanov filter

Kalmanov filter sa používa na predpovedanie stavu systémov v nadchádzajúcich intervaloch, keď ho nemožno merať priamo. Systém, ktorý bol implementovaný v publikácii [6] je lineárny, takže systémová rovnica bola definovaná takto:

$$x_k = Ax_{k-1} + w_{k-1}$$

Kde w_{k-1} je biely šum (tiež nazývaný ako procesný šum), ktorý sa vyskytuje v dôsledku nepresnosti merania aktualizovanej polohy vozidla a výpočtových úloh vykonaných počítačom, A je matica $n \times n$ definovaná ako matica prechodu stavu a $k-1$ predstavuje predchádzajúci stav.

2.1.6 Implementácia a výsledky počítania

Algoritmus bol implementovaný pomocou softvéru MATLAB. Implementácia vyššie diskutovaného algoritmu prináša výsledky s efektívnym sledovaním a detekciou vozidiel vo videu. Náklady na priradenie detekcií pomocou maďarského algoritmu vedú k presnému spočítaniu vozidiel vo videu. Týmto algoritmom sú detegované, sledované a počítané iba vozidlá vstupujúce do oblasti záujmu (detekčné pole).

2.2 Algoritmus na počítanie a klasifikáciu vozidiel

Výklad tohto algoritmu vychádza z publikácie Vehicle Counting and Classification from a Traffic Scene z roku 2008 [2]. Táto práca bola vykonaná s cieľom určiť výkon techník spracovania obrazu v klasifikácii a sčítaní pohybujúcich sa vozidiel vo videozáznamoch dopravnej scény zaznamenané stacionárnou kamerou. Boli pri nej sa použité nasledovné kroky.

2.2.1 Detekcia pozadia

Na odčítanie stabilného obrazu pozadia bola v tejto publikácii použitá adaptívna metóda detekcie pozadia. Táto metóda používa na detekciu pozadia strednú hodnotu pixelov v rozmedzí rámcov. Ak je rozptyl daného pixelu pod vopred stanovenou

prahovou hodnotou, potom sa pixel považuje za stabilný. Obrázok na pozadí sa aktualizuje podľa nižšie uvedenej rovnice.

$$B_{t+1} = B_t + S_t \times M_t$$

Tu je B_t starý obrázok na pozadí, S_t je maska, ktorá sa pohybuje medzi 0 a 1 v závislosti od rozptylu a M_t je stredná hodnota pixelov.



Obrázok 5 - Stabilné pozadie



Obrázok 6 – Skúmaný obraz

2.2.2 Kalibrácia kamery

Kalibrácia kamery zohráva dôležitú úlohu v identifikačnom procese. Kalibrácia kamery sa robí, aby sa transformovali súradnice obrazu na svetové súradnice. To je pre klasifikáciu vozidla nevyhnutné. Je to tiež veľmi dôležité, ak si niekto želá extrahovať rýchlosť vozidiel. V tejto publikácii sa brali nasledujúce predpoklady.

- Cesta je rovná (smer)
- Cesta je rovná (plochá)
- Os X cestného priestoru je kolmá na dopravný tok a os Y je s dopravným tokom rovnobežná.

Transformácia, ktorá sa uskutočnila z obrazového priestoru do svetového priestoru je projekčná transformácia. Predpokladajme, že šírka a dĺžka vybranej oblasti v cestnom priestore (ktoré sú známe) sú w a h . Rovnica na nájdenie matice projekčnej transformácie, ktorá mapuje súradnice obrazového priestoru ($ax, ay; bx, by; cx, cy; dx, dy$) na ich príslušné súradnice cestného priestoru je;

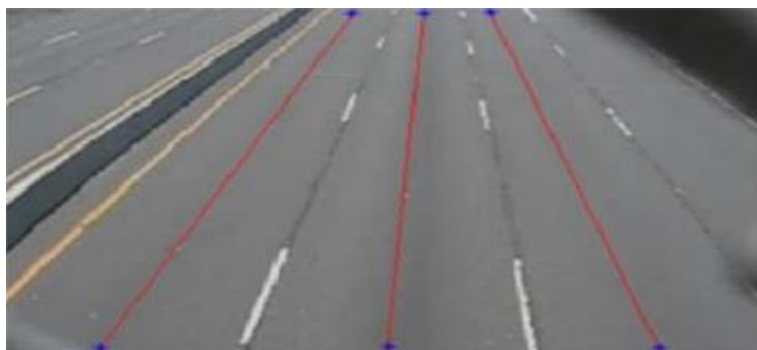
$$\begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} \begin{bmatrix} ax & bx & cx & dx \\ ay & by & cy & dy \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & w & w & 0 \\ 0 & 0 & h & h \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Na optimalizáciu výsledkov matice projekčnej transformácie, bola použitá metóda optimalizácie najmenších štvorcov. Projekčná transformácia bola použitá ako počiatočný odhad a najlepšia projekčná matica bola nájdené pomocou nasledujúcej objektívnej funkcie.

$$\sum_{i=1}^4 \left(\frac{Axi + Byi + C}{Gxi + Hyi + 1} - ui \right)^2 + \left(\frac{Dxi + Eyi + F}{Gxi + Hyi + 1} - Vi \right)^2$$

2.2.3 Kalibrácia pruhov

Cieľom publikácie je identifikovať vozidlá v každom jazdnom pruhu osobitne, takže jazdné pruhy musia byť definované a os každého jazdného pruhu musí byť identifikovaná. Užívateľ má povolený výber priestoru záujmu najskôr výberom stredu jazdných pruhov a potom východiskového a konečného bodu na každom jazdnom pruhu na definovanie sledovacej oblasti. Súradnice v priestore obrázku sa najskôr transformovali na súradnice cestného priestoru a potom sa premietli na čiary jazdného pruhu.

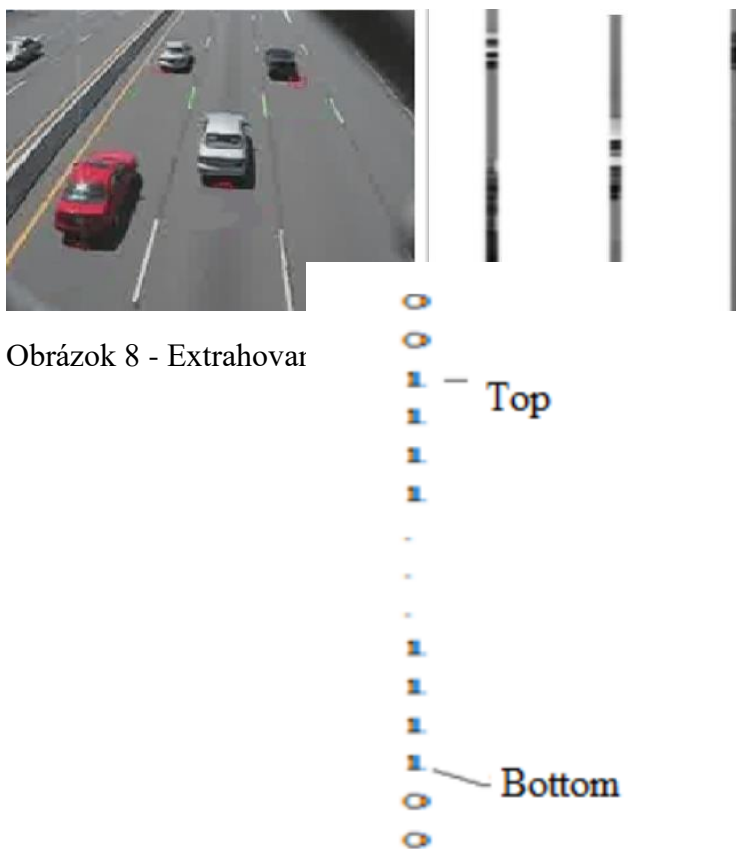


Obrázok 7 - Vybraté stredové čiary na každom pruhu

2.2.4 Sledovanie vozidiel

Na sledovanie pohybujúcich sa objektov v sledovaných oblastiach bolo vyvinutých mnoho techník ako je sledovanie bodov, sledovanie ťažísk, sledovanie obdĺžnikov atď. V tejto publikácii boli použité spodné súradnice identifikovaných predmetov na sledovanie pohybujúcich sa vozidiel vo vybranej oblasti.

Pri sledovaní novej snímky je preskúmaný začiatok stredovej čiary oblasti záujmu



Obrázok 8 - Extrahovanie

Obrázok 9 - Binárna
hodnota pixelu stĺpcového
vektora identifikovaného
vozidla

a ak sa pozoruje kladná hodnota (v tomto prípade číslo 1), algoritmus to považuje za spodnú súradnicu vozidla. Ak bola pozitívna hodnota zaznamenaná na predchádzajúcej snímke a teraz sa zmenila, naznačuje to, že vozidlo je vo vybranej oblasti záujmu v

pohybe. Detegovaním stĺpcového vektora s pozitívnym tokom hodnôt zhora nadol bolo možné detegovať vozidlo.

Vypočítaním dĺžky kladného vektorového stĺpca je možné overiť minimálnu vektorovú dĺžku vozidla. Vozidlá, ktoré nemajú minimálnu dĺžku boli zamietnuté. Ak sa nájde nové vozidlo, pridá sa do poľa vozidiel v jazdnom pruhu.

Pre každý vektor v poli vozidiel sa spustí algoritmus od poslednej známej polohy spodnej časti vektora a hľadá sa pozdĺž stredovej čiary, aby našiel ďalšiu pozíciu vektora v nasledujúcom rámci. Ak hľadanie skončilo na konci regiónu, vozidlo sa považuje za opustené zo záujmového regiónu. Tento proces sa opakuje pre všetky definované stredové čiary.

2.2.5 Klasifikácia vozidiel

Klasifikáciu v publikácii vykonali kategorizáciou vozidiel do troch tried podľa veľkosti vozidla, menovite veľké, stredné a malé. Keďže je potom ľahké nájsť dĺžku vektorov, tak sa zobrala ako parameter na klasifikáciu vozidiel podľa definovaných veľkostí. Nasledujúca tabuľka ukazuje klasifikáciu použitú v tejto publikácii.

Veľké	Autobusy, nákladiaky...
Stredné	Autá, dodávky...
Malé	Motorky, jednostopové vozidlá...

Pre každé nové vozidlo, ktoré vstúpi do jazdného pruhu v oblasti záujmu bola dĺžka vektora vypočítaná a ak vozidlo splnilo požiadavku minimálnej dĺžky, tak sa vykonala jeho klasifikácia.

2.2.6 Výsledky počítania

Výsledok závisel od mnohých faktorov ako umiestnenie kamery priamo nad cestu, kde bola premávka, poskytnutí správnych jazdných pruhov. Uskutočňovali zábery, kde sa autá približujú ku kamere a také, kde autá jazdia smerom od kamery. Systém bol skúšaný viacerými kalibráciami kamery, aby sa našiel optimálny záber. Videozáznamy

boli podrobené viacerým nezávislým testom. Samotné vozidlá sa v každom teste počítali jednak manuálne a taktiež aj pomocou systému.

Väčšina testov mala rozdiel medzi manuálnym počítaním a systémovým do 10%, avšak chybovosť sa zvýšila, keď boli pruhy z pohľadu kamery skosené, keď vozidlá nešli vo vybraných jazdných pruhoch ale aj keď išlo nejaké vyššie vozidlo, ako napríklad kamión a zakryl viac ako jeden jazdný pruh.

2.3 Algoritmus detekcie pohybujúcich sa vozidiel vo videu

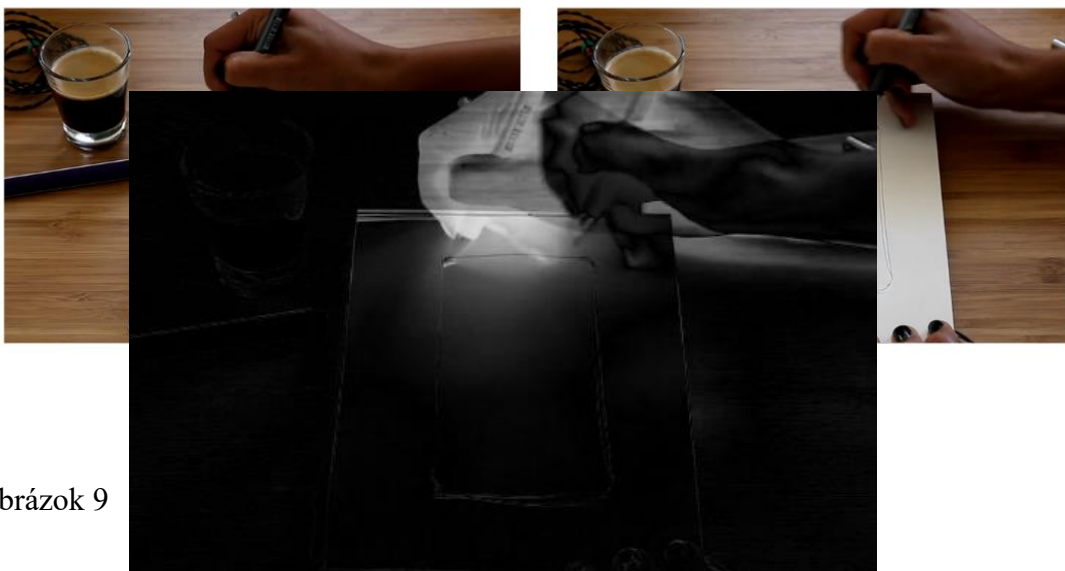
Výklad tohto algoritmu vychádza z článku Build your own Vehicle Detection Model using OpenCV and Python z roku 2020 [3]. Článok ukazuje, ako sa dá v premávke pomocou kamery naprogramovať automatický detektor a counter vozidiel v premávke. Použili v ňom nasledujúce metódy a na implementáciu technológie ako Python a knižnicu OpenCV.

2.3.1 Rozlišovanie snímok

Video je množina rámcov naskladané dokopy do správnej postupnosti. Takže, keď je vidno objekt pohybujúci sa vo videu, znamená to, že objekt je na každej nasledujúcej snímke na inom mieste. Ak sa vychádza z toho, že okrem tohto objektu sa nič iné nepohybovalo v dvoch po sebe nasledujúcich snímkach, potom rozdiel pixelov prvej a

druhej snímky zvýrazní pixely pohybujúceho sa objektu. Potom sú známe pixely a súradnice pohybujúceho sa objektu. Napríklad:

Na obrázku 9 vidno, že sa pohla iba ruka, ktorá kreslí na papier. Takže, ako bolo



Obrázok 9

Obrázok 10

spomínané vyššie, na lokalizáciu pohybujúceho sa objektu sa použije rozlišovanie snímok a výsledok vyzerá ako na obrázku 10.

2.3.2 Prahovanie

V tejto metóde je pixelovým hodnotám obrazu v odtieňoch sivej priradená jedna z dvoch hodnôt predstavujúcich čiernobiely farby na základe prahovej hodnoty. Ak je teda



Obrázok 11 - prahovanie

hodnota pixelu väčšia ako prahová hodnota, priradí sa mu jedna hodnota, inak sa priradí iná hodnota.

2.3.3 Hľadanie kontúr

Kontúry (obrysy) sa používajú na identifikáciu tvaru oblasti v obraze, ktorá má rovnakú farbu alebo intenzitu. Obrysy sú ako hranice okolo záujmových oblastí.



Obrázok 12

2.3.4 Dilatácia obrazu

Je to konvolučná operácia na obraze, kde jadro (matica) prechádza cez celý obraz. V tomto momente v článku použili dilatáciu obrazu a hneď potom opäť hľadali kontúry. Ako je vidno na obrázku 13, veľa fragmentovaných oblastí sa spojilo do seba.

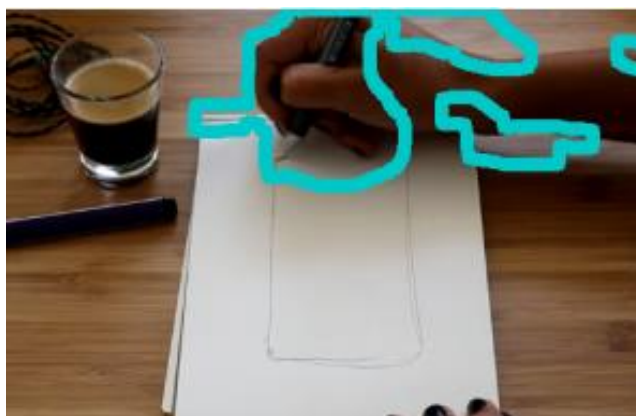


Obrázok 12



Obrázok 13

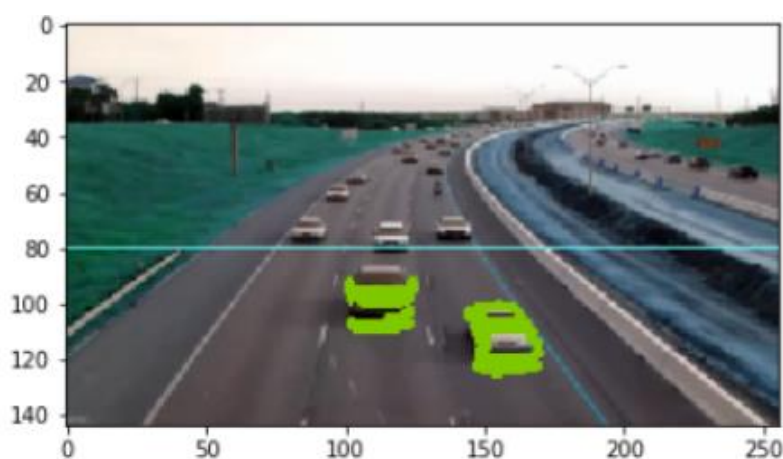
Získali takto 4 obrisy útvaru, z ktorých vybrali ten s najväčšou plochou. Keď tieto obrisy prekreslili na pôvodný obraz, vyzeralo to nasledovne.



Obrázok 14

2.3.5 Počítanie

Na počítanie použili vyššie uvedené metódy. Najprv rozlíšili viaceré snímky videozáznamu, kde detegovali pohybujúce sa pixely. Všetko, čo sa nehýbalo bolo odčítané. Na objekty v tomto kroku použili dilatáciu a hneď na to našli kontúry. Keď už mali rozoznané vozidlá od pozadia a iných irelevantných objektov, zvolili si oblasť záujmu, implementovali horizontálnu čiaru, pod ktorou sa nachádzala táto oblasť. Toto sa nakoniec použilo na pôvodný videozáznam a systém zaznamenával a počítal iba tie vozidlá, ktoré vošli do oblasti záujmu.



Obrázok 15 – Výsledok

2.4 Bakalárska práca Detekcia voľných parkovacích miest na ulici v meste [4]

Podobnej problematike sa venoval Martin Bernát v roku 2018 vo svojej bakalárskej práci. Rozdiel medzi touto bakalárskou prácou a bakalárskou prácou pána Bernáta je v tom, že jeho úlohou bolo preskúmať možnosť určovania voľných parkovacích miest s využitím snímok kamery na stĺpe verejného osvetlenia a prakticky ju overiť. úlohou tejto práce však nie je hľadať parkovacie miesta a ani ich obsadenosť. Úlohou tejto práce je rozoznať vozidlo od človeka, bicykla a iných objektov, ktoré nevchádzajú na parkovisko s účelom parkovať.

3 LITERATÚRA

- [1] Vehicle Detection, Tracking and Counting.
https://www.researchgate.net/publication/330254555_Vehicle_Detection_Tracking_and_Counting
- [2] Vehicle Counting and Classification from a Traffic Scene.
https://www.researchgate.net/publication/234136865_Vehicle_Counting_and_Classification_from_a_Traffic_Scene
- [3] Building own Vehicle Detection Model using OpenCV and Python.
<https://www.analyticsvidhya.com/blog/2020/04/vehicle-detection-opencv-python/>
- [4] Detekcia voľných parkovacích miest na ulici v meste.
<https://matobernat.github.io/Bakalarska-Praca/>