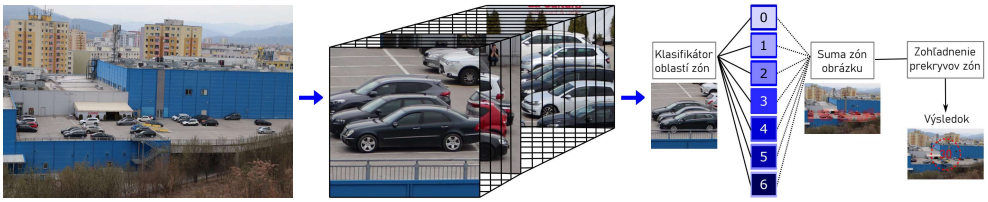
BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2021
http://excel.fit.vutbr.cz

Určenie obsadenosti parkoviska z obrazu

Pavol Dubovec*



Abstrakt

Cieľom tejto práce je vytvoriť aplikáciu, ktorá z obrazu zistí počet vozidiel na zvolenej fotografii. Takéto zisťovanie bude prebiehať klasifikáciou, pomocou konvolučnej neurónovej siete. Trénovacie dátach pozostávajú z fotografii parkovísk z rôznych pohľadov a pozícií. Riešenie bolo navrhnuté tak, že sa obrázky s parkoviskom rozdelia na niekoľko záujmových oblastí a z týchto oblastí sa vytvoria výrezy, pomocou vytvorenej aplikácie špecializovanej na túto úlohu. Následne prebehne anotácia obrázkov vytvorených týmto spôsobom, pomocou vytvorenej hodnotiacej aplikácie. Obrázky sa následne naformátujú na rovnakú veľkosť. Tieto pripravené výrezy sú následne predané knižnici na pre prácu s ML Tensorflow, pomocou ktorej prebieha trénovanie modelu. Cieľom bolo vytvoriť model, ktorý by bol dostatočne univerzálny natoľko, aby vedel určiť počet vozidiel na fotografii v akomkoľvek prostredí (čas, počasie, poveternostné podmienky) a v čo najkratšom čase. V súčasnosti model dokáže predikovať správny počet vozidiel na výrezu na testovacích dátach s presnosťou 87 % a s pripustením chyby prvého rádu na 95 %. Primárnym cieľom tejto práce je riešenie tohto problému v reálnom čase. Jedná sa klasifikáciu do 7 tried (0-6 vozidiel). Toto riešenie by mohlo byť zaujímavé hlavne pre statické kamery na netypických miestach (napr. bočný pohľad), prípadne je pre ne dôležité snímanie určitých špeciálnych oblastí parkovísk.

Kľúčové slová: Počítanie objektov — Počítanie vozidiel — Tvorba aplikácií — Deep Learning

Priložené materiály: *N/A*

*xdubov02@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Úvod

Určovanie počtu vozidiel z obrazu môže byť využité na mnohé úlohy, často použiteľné aj na riešenie praktických problémov. Práca sa zameriava najmä na problém s umiestnením kamier na parkoviskách. V súčasnosti sú mnohé parkoviská vybudované na strechách, či na miestach, kde nie je možné umiestniť kameru, tak že pohľad na jednotlivé vozidlá mali na fotografii rovnakú veľkosť a boli by zobrazené čelnou stranou vozidla, prípade z vtáčieho pohľadu. V prípade fotografií nespĺňajúcich tieto požiadavky, môže byť časovo náročné získať uspokojivý výsledok, z dôvodu nutnosti riešenia problémov, spôsobených absenciou týchto požiadaviek, ako napríklad vysoká miera prekryvu, prípadne absencia deliacich čiar. Riešenie tejto práce sa preto zameriava odlišný prístup. Fotografiu delí na zóny, v ktorých sa nachádzajú vozidlá. Týmto spôsobom je možné zjednodušiť komplexný problém, na problém klasifikácie do siedmych tried (0-6 vozidiel na obrázku), ktorý je rádovo jednoduchší na spracovanie a časovo zvládnuteľný v jednotkách stotín sekúnd.

Riešením problému je klasifikácia výrezov, vytvorených špecializovanou aplikáciou, získaných z obrázkov parkovísk. Trénovanie modelu prebieha na základe ohodnotených obrázkov, pomocou vytvorenej hodnotiacej aplikácie. Hodnotenie funkčnosti je zalo-

žené na dvoch hlavných parametroch. Prvým z nich je presnosť riešenia. Rozlišujeme dve typy presnosti. Prvou je presnosť úplná, ktorá označuje, že počet vozidiel na hodnotenom výreze bol modelom určený správne. Druhou presnosťou je presnosť s povolením chyby prvého rádu. Jedná sa o presnosť, pri ktorej pripúšťame chybu modelu, nie však väčšiu ako jedna. Druhým parametrom je rýchlosť hodnotenia počtu vozidiel na obrázku, na základe modelu. Za uspokojivý výsledok rýchlosti klasifikácie zóny je možné považovať určenie v stotinách sekúnd.

2. Teoretické základy

Problémom obsadenosti parkoviska sa zaoberá mnoho štúdií. Existujúce PGI (Parking guidance and information) systémy rozdeľujeme do štyroch kategórií založených na metóde detekcie [1]:

1. **protipólové systémy**,
2. **drôtové senzorové systémy**,
3. **systémy založené na bezdrôtových magnetických snímačoch**,
4. **systémy založené na obraze alebo kamere**.

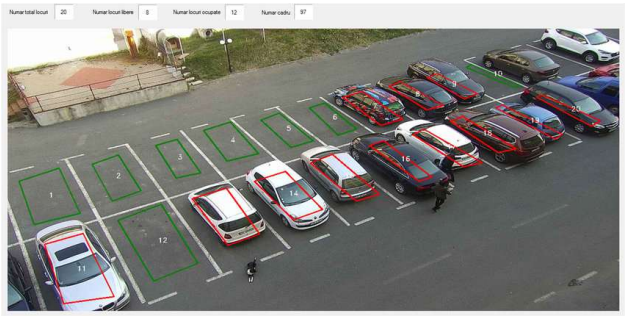
Záujmovou časťou tejto práce je spracovanie vizuálnych dát, je nutné zamerať pozornosť na túto oblasť. Systémy založené na obraze alebo kamere môžme rozdeliť podľa objektu záujmu na:

- **Car-driven algoritmy** – objektom záujmu týchto algoritmov je vozidlo. Problémom týchto riešení, je fakt, že vozidlá sa na obrázkoch alebo na snímkach z kamery nenachádzajú vždy v ideálnych podmienkach. Vozidlá sa teda môžu nachádzať v rôznych vzdialenostiach od kamery (rovnaké vozidlo rôzny počet pixelov) a v rôznych uhloch, čo komplikuje ich detekciu.
- **Space-driven algoritmy** – objektom záujmu týchto algoritmov je parkovacie miesto. Často využívajú metódu odčítania pozadia. Najväčším problémom tejto metódy je náchylnosť na zmeny pozadia (počasie, svetlo), ktoré spôsobujú jej zvýšenú chybovosť. Preto sú najvhodnejším miestom využitia týchto metód parkoviská umiestnené v interiéri.

Súčasným trendom je ale kombinácia oboch prístupov.

Niekoľko prístupov využíva riešenia pomocou mapovania parkoviska [2]. Jedná sa o činnosť, pri ktorej algoritmu určíme polohu parkovacieho miesta pomocou súradníc. Mnohokrát sa jedná o ohraničujúci útvar nachádzajúci sa vo vnútri parkovacieho miesta, obvykle obdĺžnikového alebo lichobežníkového tvaru. Spracujú sa obrázky pomocou extrakcie niekoľkých kľúčových prvkov získaných z prvotnej konfigurácie mapy. Nasleduje úprava získaných obrázkov parkovacích miest, pomocou odstránenia šumu a vyhladenia rohov pomocou na to určených špeciálnych algoritmov. Potom adaptívne odstránenie pozadia a následná úprava perspektívy. Do modelu vstupujú obrázky vyznačených oblastí parkovacích miest bez pozadia upravené na pravouhlé štvoruholníky. Takto pripravené dáta následne vstupujú do neurónovej siete.

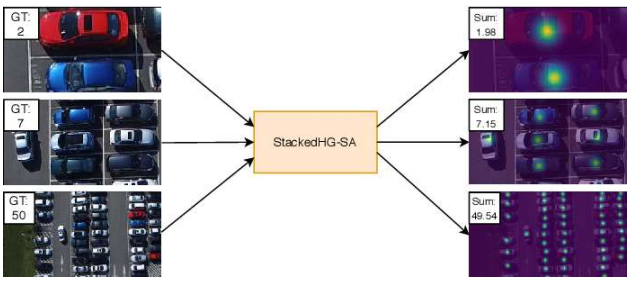
Silné stránky: Vysoká presnosť hodnotenia (93%), čiastočne umožňuje spracovanie netypických záberov
Slabé stránky: Nie vždy je možné spracovať všetky parkovacie miesta na obrázku, vyžaduje určenie súradníc parkovacích miest



Obrázok 1. Výsledná vzorka z referenčnej hodnoty 1, ktorá je súčasťou práce [2].

Iným prístupom k riešeniu problému počítania vozidiel je riešenie regresiou, pomocou hustoty [3]. Tento prístup využíva jednoduché hodnotenie vozidiel na základe bodkového značenia, kde každá bodka predstavuje jedno vozidlo. Každá bodka predstavuje matematicky deltafunkciu umiestnenú na hodnotenom vozidle. Tieto deltafunkcie sú následne konvolované s gaussovskými jadrami a následne sú výsledky skombinované tak, aby vytvorili mapu hustoty pre celý obrázok. Takto pripravené informácie následne slúžia ako základ pre neurónovú sieť (StackedHG-SA).

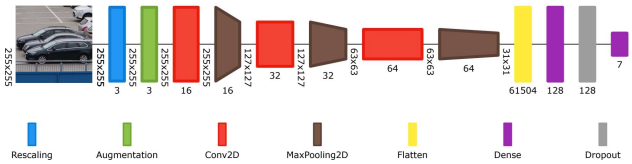
Silné stránky: Nie je nutná manuálna kalibrácia, dobre zvláda aj zmeny perspektívy a zmeny mierky.
Slabé stránky: Problém spracovania pri obrázkoch ktoré nemajú ideálne podienky (bočný pohľad, prekrývajúce sa vozidlá)



Obrázok 2. Model zohľadňujúci mierku, schopný automaticky hodnotiť kvalitu máp hustoty z [3].

3. Teória riešenia problému obsadenosti parkoviska pomocou zón

Riešenie opisované v tejto práci pracuje s parkoviskom ako so sériou zón. Tieto zóny obsahujú minimálne 0 a maximálne 6 vozidiel. Zóny sa môžu prekrývať, tento prekryv je neskôr zohľadnený pri samotnom výpočte vozidiel na obrázku. Vstupom do neurónovej siete je obrázok zóny normalizovaný na jednotnú veľkosť (predvolená veľkosť je 255×255). Dataset s takýmito obrázkami je vstupom do neurónovej siete. Ide o klasifikáciu výrezov zón do 7 tried podľa počtu vozidiel.

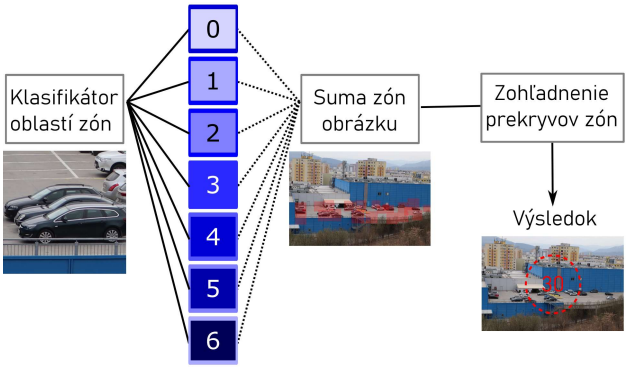


Obrázok 3. Architektúra klasifikátora zón do 7 tried. Obrázok bol čiastočne vytvorený pomocou online nástroja na automatické generovanie vizualizácií neurónových sietí z práce [4].

Architektúra neurónovej siete sa skladá z troch konvolučných vrstiev s veľkosťou jadra 3×3 a s aktivačnou funkciou ReLU. Každá konvolučná vrstva je nasledovaná maximálnou zdržovacou vrstvou (max-pooling) s veľkosťou združenia 3×3 a s veľkosťou pohybu okna združovania pri každom kroku 2×2. Po poslednom páre konvolučnej vrstvy a maxpooling vrstvy nasleduje splošťovacia (flatten) vrstva. Táto vrstva je nasledovaná plne prepojenou vrstvou so 128 jednotkami. Výpadok v sieti je nastavený na 0.5, čo vo výraznej miere zabraňuje pretrénovaniu siete. Poslednou vrstvou architektúry je plne prepojená vrstva so siedmimi jednotkami. Pred konvolučnými vrstvami sa nachádzajú vrstvy pre zmenu mierky (rescaling) na interval [0,1] a vrstva pre rozšírenie dát, zabraňujúca pretrénovaniu. Vizualizáciu tejto architektúry je možné vidieť na obrázku 3.

Vytvorenie zón z obrázku je poloautomatické. Aplikácia na vytváranie výrezov zón, ktorej činnosť je dôkladnejšie popísaná v sekcií 5, dokáže vytvoriť výrezy zón, avšak používateľ musí zvoliť, ktoré oblasti z parkoviska sú preňho zaujímavé. Toto riešenie zabezpečuje, že kamera nemusí byť umiestnená pod konkrétnym uhlom, prípadne to, že nie je nutné vidieť deliace čiary. Toto zvolenie zón, v prípade, že sa jedná o snímky zo statickej kamery, prípadne fotoaparátu je nutné vykonať len raz. Aplikácia si zapamätá súradnice zvolených zón, určí a uloží zóny aj pre ostatné obrázky snímané z rovnakého miesta.

V súčasnom stave dokáže sieť určiť triedu pre 87 % vozidiel vo veľmi krátkom čase (v jednotkách



Obrázok 4. Zóny z fotografie sú klasifikovaný do jednej zo 7 tried následne sú sčítané a v prípade, že existujú zóny, ktoré sa prekrývajú je aplikovaná korekcia týchto prekryvov. Viac o tejto korekcií je možno nájsť v sekcii 5.

sekúnd). Ak pripustíme chybu 1. rádu, tak ide o presnosť 95,1 %. Takéto riešenie by bolo možné použiť na miestach, kde je využitie tradičných metód určovania počtu objektov na obrázku, neprodukovalo dostatočné výsledky, prípadne by ich použitie v danej situácií nebolo možné (bočný pohľad, vysoká miera prekrývajúcich sa vozidiel).

4. Dataset a jeho zohľadnenie pri návrhu

Dataset pre tréning siete bol vytvorený z vlastných obrázkov a z obrázkov, ktoré mi boli poskytnuté vedúcim práce. Obrázky sú fotografované v rôzne hodiny dňa a v časti datasetu sa nachádzajú aj obrázky so zníženou viditeľnosťou (hmla, prehánky, noc).

Na parkoviskách zachytených na využitých datasetoch nenastáva často situácia, kedy by v zóne nenachádzalo žiadne auto (zóna s kapacitou 6 je prázdna), prípadne že by zóna bola plná (hraničná hodnotu veľkosti zóny). Výsledný počet výrezov s hodnotením 0 a 6 je teda v datasete zastúpený v disproporčne menšom počte ako v prípade ostatných hodnotení, znázornené na obrázku 5. V dôsledku tohto problému bola implementovaná triedne vyvážená funkcia straty [5] (class-balanced loss function) softmax. Triedovo vyvážená strata je určená na riešenie problému trénovania z nevyvážených dát zavedením váhového faktoru, ktorý je nepriamo úmerný efektívnemu počtu vzoriek. Nech n_i je počet vzoriek pre triedu i. Je nutné vypočítať efektívny počet vzoriek pre triedu i, ktorý je vyjadrený rovnicou

$$E_{n_i} = \frac{1-\beta_i^{n_i}}{1-\beta_i}, \quad \text{kde} \tag{1}$$

$$\beta_i = \frac{N_i - 1}{N_i}. \tag{2}$$

Bez bližších špecifikácií dát je veľmi ťažké empiricky

zvoliť množinu hyperparametrov N_i, pre všetky triedy. Preto v praxi predpokladáme, že N_i je viazané datasetom a množina $N_i = N$, $\beta_i = \beta = (N-1)/N$. Takže triedne vyrovnanú stratovú funkciu môžme zapísať ako

$$CB(p,y) = \frac{1-\beta}{1-\beta^{n_y}} \mathscr{L}(p,y), \tag{3}$$

kde n_y je počet prvkov v ground-truth triede y a p sú odhadované pravdepodobnosti tried modelu a $\mathscr{L}(p,y)$ je stratová funkcia modelu.

Predpokladajme predpovedané výstupy z modelu pre všetky triedy $z = [z_1, z_2, ..., z_C]^\top$, kde C je celkový počet tried. Funkcia softmax vyžaduje vzájomnú výlučnosť tried. Pravdepodobnostnou funkciou vypočítanou cez všetky triedy vypočítame ako

$$p_i = \frac{\exp(z_i)}{\sum_{j=1}^{C} \exp(z_j)}. \tag{4}$$

Predpokladajme vzorku s triednym štítkom y, tak je strata softmax cross-entropy (CE) pre túto vzorku zapísaná ako

$$CB_{softmax}(z,y) = -\log\left(\frac{\exp(z_y)}{\sum_{j=1}^{C} \exp(z_j)}\right). \tag{5}$$

Takže pre triedne vyváženú softmax cross-entropy stratovú funkciu platí vzťah

$$CB_{softmax}(z,y) = -\frac{1-\beta}{1-\beta^{n_i}} \log\left(\frac{(\exp(z_y)}{\sum_{j=1}^{C} \exp(z_j)}\right). \tag{6}$$

V ďalšom kroku sa výstup stratovej funkcie potom upraví. Úprava sa vykonáva z dôvodu, aby bolo možné urýchliť učenie siete. V tomto prípade ide o cieľ, kedy je potrebné, aby sa model nemýlil o niekoľko tried, teda chyba modelu nebola väčšia ako 1. Táto úprava je vykonaná pomocou multiplikátora, ktorým sa vynásobí strata vypočítaná pomocou triedne vyváženej softmax cross-entropy stratovej funkcie. Multiplikátor je vypočítaný ako

$$m = e^{(|p-t|)}, \tag{7}$$

kde m je multiplikátor stratovej funkcie, e je Eulerovo číslo, p je predpokladaná hodnota, t je pravdivá hodnota. Výsledná strata je vypočítaná ako

$$CB_{WS}(z,y) = \left[-\frac{1-\beta}{1-\beta^{n_i}} \log\left(\frac{\exp(z_y)}{\sum_{j=1}^{C} \exp(z_j)}\right)\right] \times m. \tag{8}$$

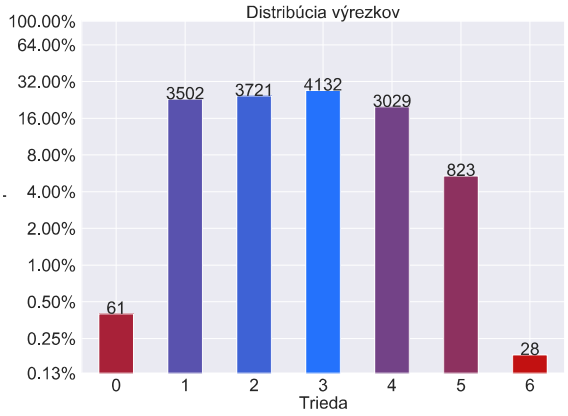
5. Implementácia

Implementácia sa skladá zo 4 častí:

1. Aplikácia na orezávanie obrázkov.
2. Aplikácia na anotáciu, normalizáciu a porovnávanie výsledkov.
3. Trénovanie modelu a pokusy.

V prvej časti implementácie bolo nutné vytvoriť aplikáciu, ktorá z fotografie parkoviska ľubovoľnej veľkosti vytvorí niekoľko výrezov zvolených zón. Zóna je oblasť tvaru štvorca ľubovoľnej veľkosti, za podmienky, že sa v každej zóne môže nachádzať x počet vozidiel, kde $x \in \{0,1,2,3,4,5,6\}$. Užívateľ má možnosť zvoliť si novú zónu prostredníctvom dvoch nezávislých kliknutí na obrázok s fotografiou. Kliknutím ľavého tlačidla myši používateľ zvolí prvý bod a kliknutím pravého tlačidla druhý. Po kliknutí na tlačidlo *(Pridať štvorec)* vznikne medzi zvolenými bodmi štvorec (Ak používateľ zvolil obdĺžnik tak sa na štvorec dopočíta). Štvorec je možné ľavým dvojklikom označiť a presunúť na iné miesto, odstrániť tlačidlom *(Odstrániť štvorec)*, zväčšiť tlačidlom *(+)* a zmenšiť tlačidlom *(-)*. Všetky operácie s úpravou štvorca je nutné potvrdiť dvojklikom pravého tlačidla myši pre uloženie úprav. Aplikácia umožňuje načítanie zón z minulej relácie, výpis dát do súboru bez samotného orezania, zvolenie názvu datasetu a vykonanie orezávania nad všetkými fotografiami v zložke (predpokladá dataset). Výstupom aplikácie sú výrezy z obrázku a 3 JSON súbory (koordináty štvorcov, susedné štvorce a ich spoločná plocha, názvy rodičovských a dcérskych fotografií).

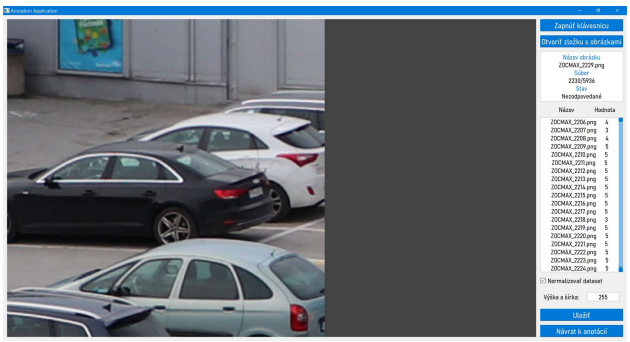
Ďalšou časťou implementácie je vytvorenie aplikácie na hodnotenie výrezov zón. Aplikácia umožňuje hodnotenie výrezu hodnotami od 0 po 6 a špeciálnu hodnotu *(?)* znázorňujúcu chybný obrázok datasetu. Je taktiež umožnená úprava hodnotenia už ohodnotených výrezov. Je možné prehliadanie už ohodnotených výrezov pomocou šípok. Ďalšou funkciou aplikácie



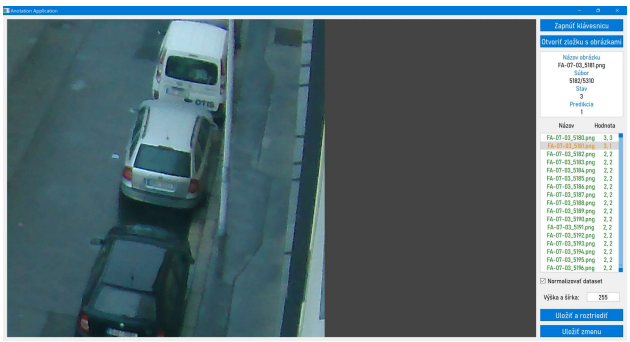
Obrázok 5. Počet výrezov v jednotlivých triedach znázornený na logaritmickom meradle.

Obrázok 6. Ukážka vytvárania zón v aplikácii na orezávanie fotografií.

je normalizácia výrezov na jednotnú veľkosť a ich roztriedenie do prislúchajúcich triednych zložiek. Takto spracované výrezy je možno priamo načítať ako dataset API Keras. Aplikácia taktiež môže slúžiť na porovnávanie predikovaného výstupu s pravdivým, ak aplikácií poskytneme súbor s predikciami, ktorý je výstupom z hodnotenia testovacieho datasetu prostredníctvom vytrénovaného modelu.



Obrázok 7. Ukážka hodnotiacej aplikácie v režime hodnotenia obrázkov. Aplikácia umožňuje úplne ovládanie myšou, prípadne prostredníctvom klávesnice.



Obrázok 8. Ukážka hodnotiacej aplikácie v režime porovnávania s výstupom modelu nad testovacími dátami.

Trénovanie modelu prebiehalo v API Keras. Dataset je načítaný prostredníctvom čítania zo zložky. Sú vytvorené 3 datasety: trénovací a validačný tvorený výrezmi z anotačnej aplikácie pomocou normalizácie a dataset trénovací získaný rovnakým spôsobom avšak bez triedenia. Následne je zostavený model, ktorý je možné vidieť na obrázku 3. Ďalej prebieha trénovanie, s vlastnou funkciou straty podrobne popísanej v [5]. Po trénovaní nasleduje testovanie modelu prostredníctvom testovacieho datasetu a je určená presnosť modelu a presnosť s pripustením chyby prvého rádu.

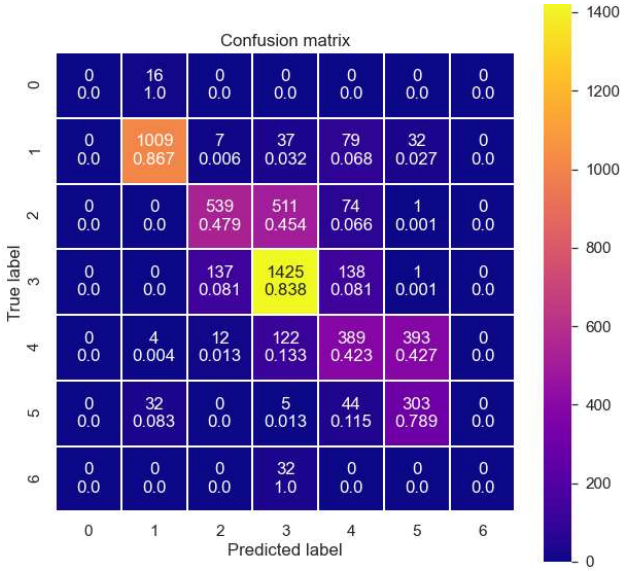
Poslednou časťou implementácie výpočet celkového počtu vozidiel na obrázku. Toto je vykonané súčtom predpokladaných hodnôt zón prislúchajúcich jednotlivým obrázkom z ktorých boli vytvorené. Čas potrebný pre evaluáciu jedného výrezku je v priemerne 0.048 s. Čas pre výpočet celkového počtu vozidiel na obrázku je priamo úmerne závislý na počte zón. Navyše je aplikovaná korekcia pre hodnoty tých výrezov zón, ktoré majú spoločnú plochu s inou zónou. Táto korekcia je vykonaná vzťahom

$$hk_{z_1} = \left(\left(1 - \frac{S_{z_1 \cap z_2}}{S_{z_1}}\right) \times h_{z_1}\right) + \left(\frac{S_{z_1 \cap z_2}}{S_{z_1}} \times h_{z_2}\right), \quad (9)$$

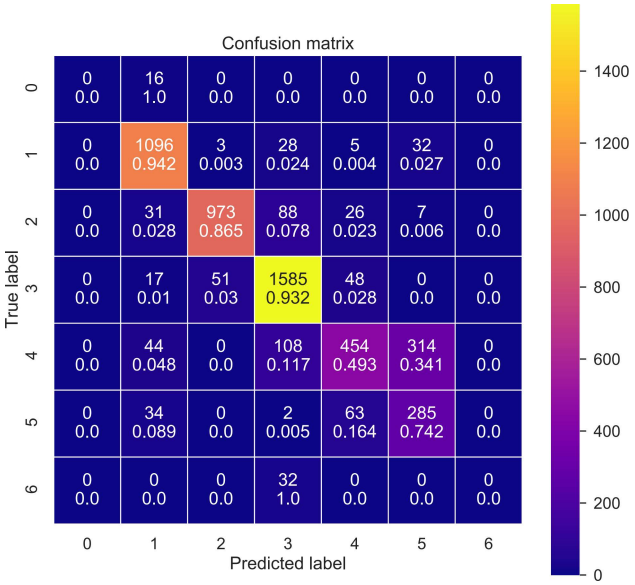
kde hk_{z_1} je počet vozidiel zóny 1 po korekcií, $\frac{S_{z_1 \cap z_2}}{S_{z_1}}$ je pravdepodobnosť výskytu v spoločnej ploche zón 1 a 2 vzhľadom na zónu 1, h_{z_1} a h_{z_2} je počet vozidiel v zóne 1 a 2, ktoré boli určené modelom.

6. Experimenty

Experimenty boli vykonávané prostredníctvom vykresľovania prípadne výpisu. Najčastejšími experimentami boli zmeny stratovej funkcie, augmentácie a hyper-parametrov modelu. Prvý typ výstupu experimentu je znázornený na obrázku 9, 10. V oboch

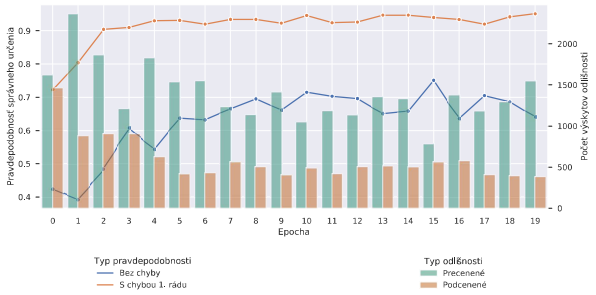


Obrázok 9. Ukážka výstupu experimentu po 18 epoche prostredníctvom matice zámen vygenerovanej pri evaluácii modelu na testovacích dátach výrezkov zón. Využitá bola loss funkcia class-balanced softmax.



Obrázok 10. Ukážka výstupu experimentu po 50 epoche prostredníctvom matice zámen vygenerovanej pri evaluácii modelu na testovacích dátach výrezkov zón. Tentokrát bola využitá funkcia straty sparse categorical cross entropy.

prípadoch sa jedná o maticu zámien, ale pri každom experimente bola využitá iná funkcia straty. Matice tohto typu sú vytvárané, po každej epoche pre kontrolu výstupu. Ďalším typom výstupu experimentu je obrázok 11, znázorňujúci testovaciu presnosť v jednotlivých epochách.



Obrázok 11. Graf znázorňujúci presnosť v jednotlivých epochách na testovacom datasete a počet odchýlok (podcenení/precenení) v rámci testovaných výrezov.

7. Záver

Cieľom tejto práce bolo vypočítať počet vozidiel na obrázku. Pre riešenie bola zvolená metóda deľby obrázku na zóny, ktoré sú následne spracované najskôr pomocou aplikácie na vytváranie výrezov a neskôr anotácia hodnotiacou aplikáciou. Model pre predikciu bol trénovaný na obrázkoch zón o jednotnej veľkosti.

Aplikácia dokáže určiť správny počet vozidiel zobrazených na obrázku s presnosťou 87 % a s pripustením chyby prvého rádu 95,1141 %. Práca využíva riešenie pomocou triedne vyrovnanej stratovej funkcie softmax, ktorej výsledok je následne násobený multiplikátorom, ktorý zabezpečí rýchlejšie učenie modelu vyššou penalizáciu snímok, ktorých očakávaný a predpokladaný výstup sa odlišujú o niekoľko tried.

Prínosom tejto práce je jednoduché, relatívne rýchle a výpočtovo nenáročné riešenie. Dôraz bol kladený hlavne na rýchlosť určenia výsledku, jednoduchosť získavania dát a účinnosť riešenia aj v prípade, že nenastávajú ideálne podmienky (parkovisko bez deliacich čiar, bočný pohľad a iné), v dôsledku čoho by praktické uplatnenie riešenia nemuselo byť veľmi finančne náročné.

Riešenie by mohlo byť využiteľné vtedy, ak parkovisko nemá ideálne parametre alebo v prípade, ak kamera umiestnená na parkovisku nezviera s vozidlami požadovaný uhol.

Poďakovanie

Ďakujem vedúcemu bakalárskej práce pánovi profesorovi Adamovi Heroutovi za pomoc a rady pri písaní tejto práce.

Literatúra

[1] H. Ichihashi, A. Notsu, K. Honda, T. Katada, and M. Fujiyoshi. Vacant parking space detector for outdoor parking lot by using surveillance camera and fcm classifier. In *2009 IEEE International Conference on Fuzzy Systems*, FUZZ-IEEE'09, page 127–134. IEEE Press, 2009.

[2] Tătulea, Călin, Brad, Remus, Brâncovean, and Greavu. An image feature-based method for parking lot occupancy. *Future Internet*, 11:169, 08 2019.

[3] P. Dobeš, J. Špaňhel, V. Bartl, R. Juránek, and A. Herout. Density-based vehicle counting with unsupervised scale selection. In *2020 Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8, 2020.

[4] Alex Bauerle, Christian Van Onzenoodt, and Timo Ropinski. Net2vis - a visual grammar for automatically generating publication-ready cnn architecture visualizations. *IEEE Transactions on Visualization and Computer Graphics*, page 1–1, 2021.

[5] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples, 2019.