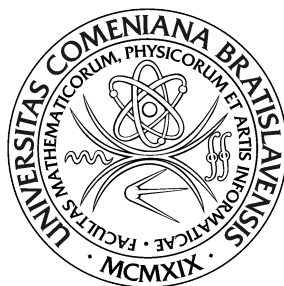


UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



DETEKCIA VIZUÁLNEHO SMOGU PRI CESTÁCH

Diplomová práca

2022

Bc. Michal Zrubec

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



DETEKCIA VIZUÁLNEHO SMOGU PRI CESTÁCH

Diplomová práca

Študijný program: Aplikovaná informatika
Študijný odbor: 2511 Aplikovaná informatika
Školiace pracovisko: Katedra aplikovanej informatiky
Školiteľ: RNDr. Zuzana Černeková, PhD.

Bratislava, 2022

Bc. Michal Zrubec

Čestné prehlásenie: Čestne prehlasujem, že som túto diplomovú prácu
vypracoval samostatne s použitím uvedenej literatúry.

V Bratislave dňa:

.....
Bc. Michal Zrubec

Pod'akovanie: Ďakujem svojej školiteľke RNDr. Zuzane Černekovej, PhD. za poskytnutý čas, poskytnutý materiál, cenné rady a pomoc pri riešení tejto práce.

Abstrakt

Cieľom tejto práce je preskúmať možnosť využitia neurónových sietí na detekciu reklamných plôch popri cestách. Detegovať významnosť reklám na základe saliency a následne verifikovať pomocou eyetrackera. V prvej kapitole sú popísané reklamné plochy a podobné práce. V druhej kapitole sú podrobnejšie popísané neurónové siete. V tretej kapitole je popísaný návrh riešenia problému a návrh aplikácie. V nadväznosti na návrh je popísaná samotná implementácia konkrétnej aplikácie na detekciu reklamných plôch a významnosti reklamných plôch.

Kľúčové slová: spracovanie obrazu, neurónové siete, významnosť, reklamné plochy

Abstract

TBD

Keywords: TBD

Obsah

1	Úvod	1
2	Prehľad problematiky	2
2.1	Reklamné plochy	2
2.2	Detekcia objektov	2
2.3	Motivácia detekcie reklamných plôch	3
2.4	Podobné práce	3
2.4.1	Detekcia billboardov	3
2.4.2	Detekcia reklám	4
2.4.3	Detekcia účastníkov cestnej premávky	5
3	Teoretická časť	8
3.1	Detekcia významných oblastí v obraze	8
3.2	Neurónové siete a umelá inteligencia	10
3.2.1	Biologický neurón	11
3.2.2	Umelý neurón	11
3.2.3	Umelé neurónové siete	12
3.2.4	Učenie neurónovej siete	13
3.2.5	Konvolučné neurónové siete	13
3.3	Hyperparametre	15

3.4	Dataset	16
3.5	Detekcia a klasifikácia objektov	17
3.6	Detekcia objektov	17
3.6.1	Rodina R-CNN (Region-based Convolutional Neural Networks)	18
3.6.2	SSD (Single Shot Detector)	20
3.6.3	YOLO (You Only Look Once)	21
4	Návrh riešenia problematiky	22
4.1	Detekcia reklamnej plochy	22
4.2	Detekcia významnosti reklamnej plochy	23
5	Riešenie problematiky a implementácia	24
5.1	YOLOv5	24
5.2	Použité nástroje	25
5.2.1	Pytorch	25
5.2.2	Weights & Biases	25
5.3	Dataset	25
5.3.1	Príprava datasetu a súborov pre tréning	26
5.4	Tréning	27
6	Vyhodnotenie a výsledky	28

Kapitola 1

Úvod

Cieľom reklamy je propagovať nejaký výrobok, informáciu, službu alebo názory so zameraním na potencionálny trh. Reklama môže mať mnoho foriem a podôb. Jednou z foriem reklamy je využívanie tzv. reklamných plôch - billboardov. V posledných desaťročiach billboardy zaplavili aj mestá, no ich hlavné zameranie je na vodičov dopravných prostriedkov. Reklamné plochy môžeme nájsť v okolí ciest a diaľníc, preto je dôležité, aby informácie na nich zobrazené boli ľahko čitateľné aj počas jazdy.

Detekcia objektov je momentálne jednou z najžiadanejších technológií počítačového videnia a spracovania obrazu. Pod týmto pojmom rozumieme určenie polohy a hraníc daného objektu v obraze. Využitie nachádza v rôznych odvetviach a aktuálne dochádza k veľkému rozmachu tejto technológie v medicíne, napríklad pri detekcii nádorov alebo obličkových kameňov, kde by postupne mohla zjednodušiť a zlepšiť prácu lekárov. Ďalším uplatnením detekcie objektov je automobilový priemysel, kde sa využíva pri vývoji autonómnych vozidiel na monitorovanie rôznych objektov v okolí auta. Môže ísť napríklad o detekciu chodcov, pruhov, dopravných značiek, atď.

TODO V prvej kapitole..

Kapitola 2

Prehľad problematiky

2.1 Reklamné plochy

Cieľom reklamy je propagovať nejaký výrobok, informáciu, službu alebo názory so zameraním na potencionálny trh. Reklama môže mať mnoho foriem a podôb. Jednou z foriem reklamy je využívanie tzv. reklamných plôch - billboardov. V posledných desaťročiach billboardy zaplavili aj mestá, no ich hlavné zameranie je na vodičov dopravných prostriedkov. Reklamné plochy môžeme nájsť v okolí ciest a diaľníc, preto je dôležité, aby informácie na nich zobrazené boli ľahko čitateľné aj počas jazdy.

2.2 Detekcia objektov

Detekcia objektov je momentálne jednou z najžiadanejších technológií počítačového videnia a spracovania obrazu. Pod týmto pojmom rozumieme určenie polohy a hraníc daného objektu v obraze. Využitie nachádza v rôznych odvetviach a aktuálne dochádza k veľkému rozmachu tejto technológie v medicíne, napríklad pri detekcii nádorov alebo obličkových kameňov, kde by postupne

mohla zjednodušiť a zlepšiť prácu lekárov. Ďalším uplatnením detekcie objektov je automobilový priemysel, kde sa využíva pri vývoji autonómnych vozidiel na monitorovanie rôznych objektov v okolí auta. Môže ísť napríklad o detekciu chodcov, pruhov, dopravných značiek, atď.

2.3 Motivácia detekcie reklamných plôch

Motiváciou tejto práce je zistiť relevantnosť a sledovanosť reklamných plôch popri cestách pomocou neurónových sietí a a eyetrackera. Výsledkom týchto zistení by mohla byť spätná väzba majiteľom alebo tvorcom reklamných plôch o sledovanosti jednotlivých reklám a prípadné odstránenie nesledovaných reklám, čo by viedlo k odstráneniu zbytočného reklamného smogu.

2.4 Podobné práce

2.4.1 Detekcia billboardov

Cieľ práce

Cieľom práce autorov bolo vytvorenie systému, ktorý má za úlohu detekciu reklamných billboardov a označovanie ich polohy pomocou hlbokaj konvolučnej neurónovej siete [RDS⁺19].

Riešenie

Pri riešení problému sa autori rozhodli použiť predtrénovaný model, ktorý používal architektúru AlexNet hlbokaj konvolučnej neurónovej siete. Sieť pozostávala z piatich konvolučných vrstiev, troch pooling vrstiev a troch plne prepojených vrstiev. Na dotrénovanie tohto predtrénovaného modelu použili

metódu transfer learning. Použili pri tom 300 obrázkov billboardov z Google a 84 obrázkov billboardov zachytených na telefóne. Označovanie polohy prebiehalo pomocou android telefónu, na ktorom použili GPS a 4G sieť pre určenie polohy.

Layer	Size	Kernel	Neuron
Input: (Resized Image)	224 x 224 x 3 x 1	-	150528
Layer 1: Convolution + Max Pool	55 x 55 x 48 x 2	96	290400
Layer 2: Convolution + Max Pool	27 x 27 x 128 x 2	256	186624
Layer 3: Convolution	13 x 13 x 192 x 2	384	64896
Layer 4: Convolution	13 x 13 x 192 x 2	384	64896
Layer 5: Convolution + Max Pool	13 x 13 x 128 x 2	256	43264
Layer 6: Fully-Connected	2048 x 2	-	4096
Layer 7: Fully-Connected	2048 x 2	-	4096
Layer 8: Softmax Output	1000 x 1	-	1000

Obr. 2.1: Ukážka použitej architektúry [RDS⁺19]

Testovanie a výsledky

Pri tréňovaní siete autori vykonali dva experimenty. Pri prvom sa vykonalo 4000 krokov tréňovania s presnosťou 89.3% a pri druhom sa vykonalo 8000 krokov tréňovania s presnosťou 92.7%. Pri testovaní systému dosiahli presnosť 72%, kde problém robili hlavne billboardy, ktoré sa nachádzali v obrázkoch ďaleko s presnosťou detekcie a určenia polohy iba 28.1%.

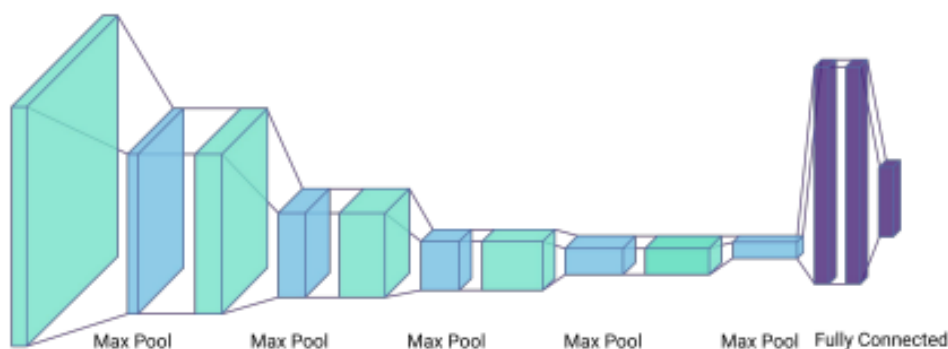
2.4.2 Detekcia reklám

Cieľ práce

Cieľom tejto práce bolo vytvoriť systém s využitím neurónovej siete, ktorý zisťuje, či sa na danej snímke videa nachádza nejaké reklamná plocha [HDN⁺18].

Riešenie

Na riešenie tohto problému autori vytvorili sieť ADNet (advert-detection model), ktorá bola inšpirovaná VGG19 architektúrou. Model tejto siete mal veľmi malé konvolučné filtre (3x3) s hĺbkou 19 vrstiev a táto sieť mala predtrénované váhy na ImageNet datasete. Autori pri riešení z tejto siete použili prvých 5 predtrénovaných vrstiev a z pôvodnej architektúry nahradili posledné 3 váhové vrstvy plne prepojenými vrstvami. Sieť bola trénovaná na skoro 19 tisíc obrázkoch, kde približne polovica obsahovala nejaké billboardy. Obrázky pochádzali z Mapillary Vistas Datasetu a MS-COCO datasetu.



Obr. 2.2: Ukážka použitej architektúry [HDN⁺18]

Testovanie a výsledky

Na trénovanie bolo použitých 13300 obrázkov a zvyšok bol použitý na testovanie. Pri testovaní bola dosiahnutá presnosť klasifikácie 94%.

2.4.3 Detekcia účastníkov cestnej premávky

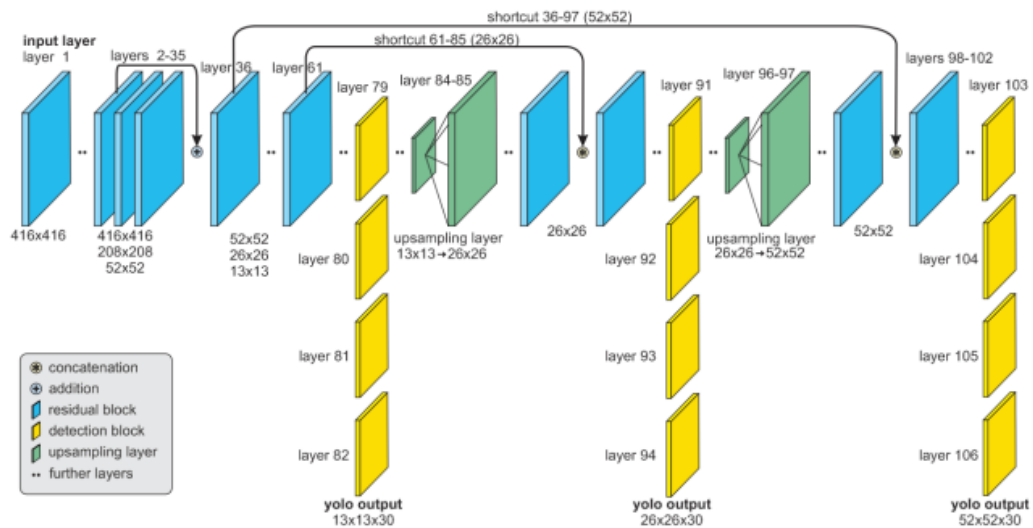
Cieľ práce

Autori sa v tomto článku zaoberali detekciou účastníkov cestnej premávky pomocou neuronových sietí. Zamerali sa pritom na klasifikáciu áut, kamiónov,

chodcov, dopravných značiek a semaforov [18].

Riešenie

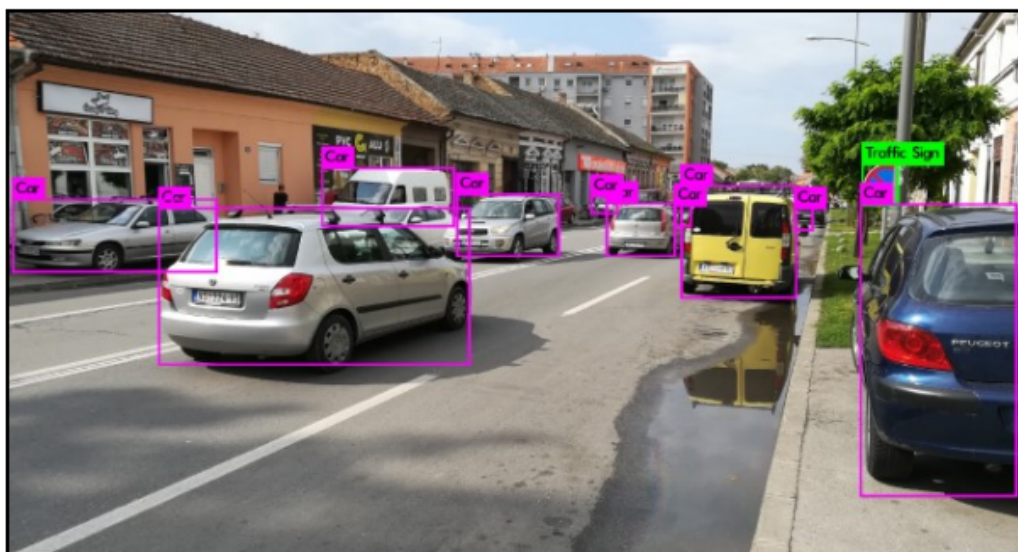
Pri riešení sa rozhodli použiť YOLOv3 architektúru. Tento model pozostáva zo 106 vrstiev. Na tréning tejto siete použili Berkeley Deep Drive dataset, ktorý pozostával zo 70000 tréningových obrázkov a 30000 validačných obrázkov a bol použitý model s predtrénovanými váhami na COCO datasete.



Obr. 2.3: Ukážka použitej architektúry [18]

Testovanie a výsledky

Po 120 epochách tréningu sa podarilo získať presnosť 63%, mAP 46% a priemernú IoU (intersect over union) 46%.



Obr. 2.4: Výsledok detekcie účastníkov cestnej premávky [18]

Kapitola 3

Teoretická časť

3.1 Detekcia významných oblastí v obraze

Pojem významná oblasť si vieme vysvetliť ako nejaké viditeľné lokálne zmeny v obraze. Pri významných oblastiach rozlišujeme dva typy významnosti - Nízko-úrovňová a vysoko-úrovňová významnosť. Pri nízko-úrovňovej je pozornosť podvedome priťahovaná k lokálnym zmenám v obraze. Pri vysoko-úrovňovej zase pozornosť reaguje na pridelené úlohy.

Vizuálna pozornosť môže mať dva prístupy:

- Zdola nahor - je to podvedomá reakcia na podnety. Ide väčšinou o jasné zmeny v obraze, ako napríklad zmena intenzity, orientácie, textúry, polohy, atď. Príkladom je napríklad červené jablko na zelenom strome alebo oči a pery na ľudskej tvári. Procesy sú nezávislé na úlohe, ktorú pozorovateľ rieši a využívajú sa pri automatickej detekcii významných oblastí v scéne.
- Zhora nadol - pozorovanie scény je čiastočne ovládané pozorovateľom, napríklad pozorovateľ je ovplyvnený vecami čo zažil v minulosti, takže

využíva svoje skúsenosti. Prídavnou informáciou je úloha, ktorú rieši. Príkladom je napríklad hľadanie nejakého konkrétného objektu, pri ktorom môže prehliadnuť iné významné oblasti v scéne.

Detekcia významných oblastí v obraze je môže byť riešená dvomi prístupmi:

- Sledovanie pohybu očí (eye tracking) - vykonáva sa to pomocou zariadenia eyetracker, ktorý sleduje pohyb očí pomocou senzorov a metód počítačového videnia.
- Model na detekciu významných oblastí.

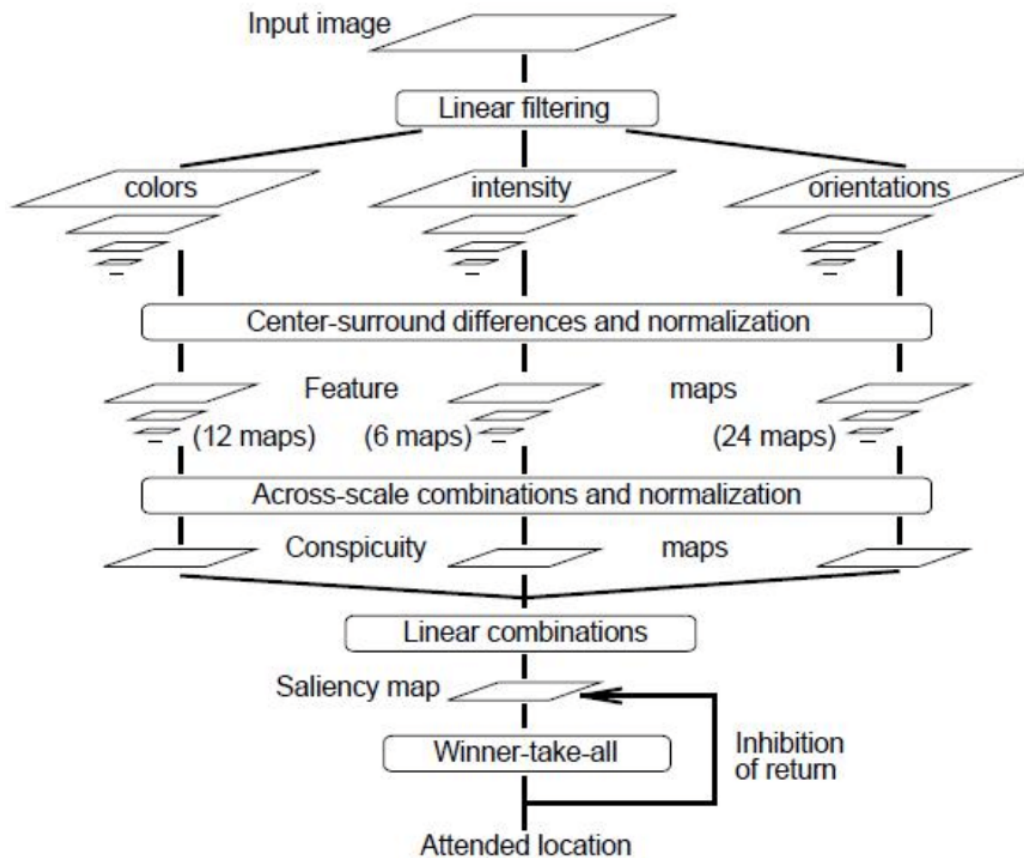


Obr. 3.1: Bežný model eye trackera

Ittiho model

Ittiho model bol navrhnutý v roku 1998 a je to najznámejší model pre výpočet významných oblastí. Model kombinuje viacúrovňové príznaky do jednej mapy významných oblastí (saliency map) a príznaky v obraze sa hľadajú v troch paralelných kanáloch pre farbu, intenzitu a orientáciu, kde sa hľadajú zmeny. Model celkovo vytvorí 42 máp, z ktorých 6 prislúcha intenzite, 12 farbe a 24

orientácii. Týchto 42 máp je následne lineárne skombinovaných do jednej saliency mapy [IKN98].



Obr. 3.2: Ukážka Ittiho modelu [Hay08].

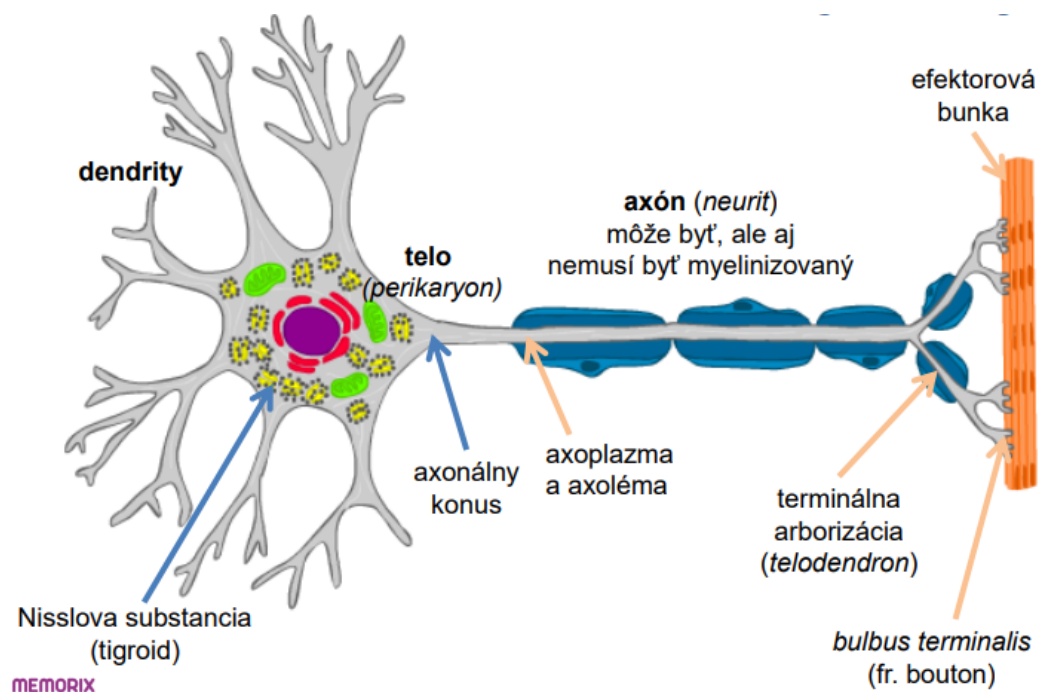
3.2 Neurónové siete a umelá inteligencia

Neurónové siete vychádzajú z predpokladu, že základným kameňom stavby ľudského mozgu je neurón.

3.2.1 Biologický neurón

Biologický neurón je nezávislá atomická a funkčná jednotka, ktorej úlohou je získavanie, prenos, spracovanie a ukladanie informácií. Základnými časťami neurónu sú:

- Neurocyt alebo perikaryón - telo neurónu,
- Dendrit - prijíma vstupnú informáciu z okolia od ostatných neurónov,
- Neurit alebo axón - vedie informáciu od tela neurónu iným neurónom do okolia.

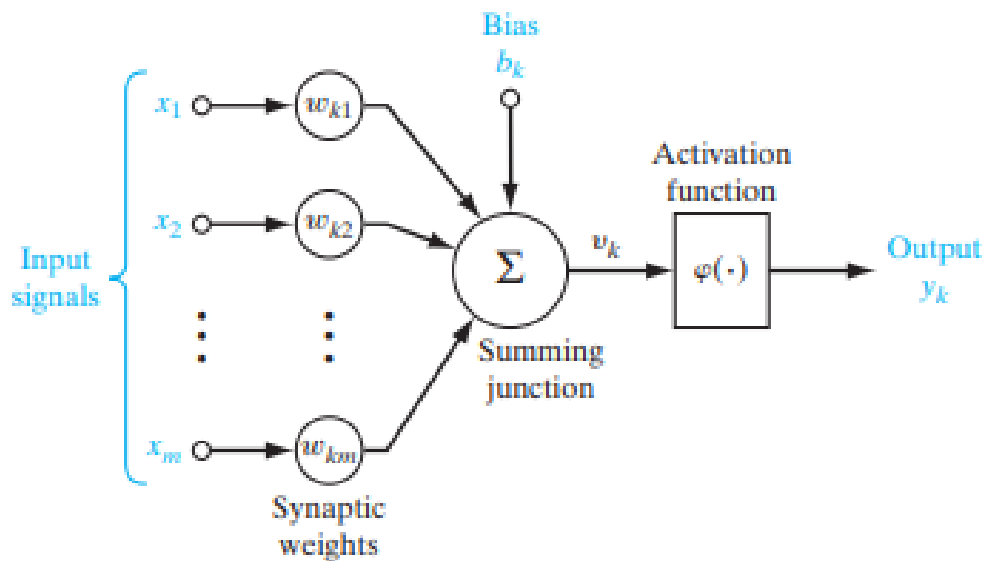


Obr. 3.3: Schéma biologického neurónu

3.2.2 Umelý neurón

Základnou funkciou umelého neurónu je zber elementárnych informácií, ich vyhodnotenie a generovanie zodpovedného výstupu. Neurón môže mať ľub-

voľný počet vstupov, ale vždy iba jeden výstup. V roku 1943 McCulloch a Pitts navrhli jeden z najjednoduchších modelov neurónu. Tento jednoduchý model logického neurónu je pokladaný za dôležitý a významný krok v rozvoji umelej inteligencie [Hay08].



Obr. 3.4: Schéma umelého neurónu [Hay08]

3.2.3 Umelé neurónové siete

Každá umelá sieť sa skladá zo vzájomne prepojených umelých neurónov, kde výstup jedného neurónu je vstupom do ďalších neurónov. Počet neurónov a ich vzájomné prepojenie určujú topológiu umelej neurónovej siete, ktorá je určená orientovaným grafom, ktorý predstavuje prepojenia medzi jednotlivými neurónmi. Z hľadiska štruktúry sú umelé neuróny organizované do vrstiev, pričom v každej vrstve sa nachádzajú neuróny so spoločnou charakteristikou [Hay08].

Každá neurónová sieť pozostáva z minimálne dvoch vrstiev - vstupná a výstupná, prípadne ďalších vrstiev, ktoré sa nazývajú aj "skryté", pretože

ich vstupy a výstupy priamo nezasahujú do vonkajšieho prostredia. Vstupná vrstva má za úlohu distribúciu vstupných signálov siete do ďalších vrstiev. Neuróny v tejto vrstve majú jeden vstup, ktorý je aj vstupom do neurónovej siete. Prenosová funkcia reprezentuje lineárnu funkciu, takže neuróny odosielať bez zmeny vstupný signál ako výstupný. Výstupná vrstva definuje výstup neurónovej siete [Hay08].

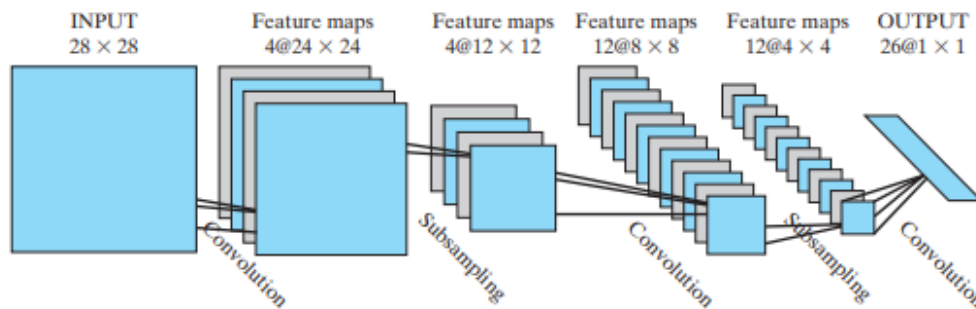
Neurónová sieť je sa dá charakterizovať na základe niekoľkých vlastností. Medzi najhlavnejšie patrí topológia siete, spôsob učenia, atď.

3.2.4 Učenie neurónovej siete

Dôležitou schopnosťou neurónových sietí je schopnosť učiť sa. Ide o algoritmus nastavenia váh, tak aby systém čo najlepšie spracovával aj neznáme informácie. Pre umelé neurónové siete rozlišujeme dva typy učenia. Ide o učenie s učiteľom a učenie bez učiteľa. Pri metóde učenia s učiteľom je známy požadovaný výstup. Pri metóde učenia bez učiteľa nie sú známe žiadne požadované výstupy a váhy sa nastavujú len pomocou vstupných údajov. Sieť potom sama rozhodne, ktorá odozva je najlepšia a sama nastaví váhy.

3.2.5 Konvolučné neurónové siete

Jedným z najpopulárnejších prístupov, ktoré sa používajú na riešenia úloh detekcie a klasifikácie objektov je použitie konvolučných neurónových sietí, ktoré dokážu využiť vstupné obrázky na naučenie sa vlastností objektov pre vyriešenie daného problému. Konvolučná neurónová sieť je špecifický typ neurónovej siete, ktorá sa skladá z jednej alebo viacerých konvolučných vrstiev, za ktorými nasleduje podobne ako v štandardných viacvrstvových neurónových sieťach jedna alebo viac plne prepojených vrstiev.



Obr. 3.5: Príklad konvolučnej neurónovej siete [Hay08]

Vrstvy konvolučných neurónových sietí:

- Vstupná vrstva - úlohou tejto vrstvy je snímať obrázok s jeho rozmermi šírky, výšky a troch kanálov - červený, modrý a zelený (pre šedotónový obrázok iba jeden kanál)
- Konvolučná vrstva - jej úlohou je zistiť lokálne informácie o rastrovom obrázku. Niekedy sa používajú aj veľmi malé oblasti veľkosti 3x3 alebo 5x5 pixelu. Táto oblasť je vstupom do jedného neurónu (teda 9 vstupov pri 3x3). Jednotlivé oblasti sa navzájom prekrývajú, čiže pre obrázok s rozmermi $x \times y$ a pre oblasti s veľkosťou $k_x \times k_y$ budeme mať spolu $(xk_x+1)(yk_y+1)$ oblastí, a teda rovnaký počet neurónov v konvolučnej vrstve. Z konvolučných vrstiev požadujeme získať n rovín, pričom n by malo zhruba zodpovedať miere informácií, ktoré v obrázku hľadáme [fra].
- Poolingová vrstva - na svojom vstupe akceptuje výstup z predchádzajúcej konvolučnej vrstvy. Táto vrstva efektívne znižuje horizontálne aj vertikálne rozlíšenie obrázku. Navyše táto vrstva vykonáva len jednoduché výpočty, napríklad výber pixelu s maximálnou hodnotou a podobne. Tým, že sa vrstvy do siete vkladajú v poradí konvolučná $\rightarrow$

poolingová $\rightarrow$ konvolučná, dochádza k postupnému zmenšovaniu množstva dát, takže vstupy do poslednej časti siete sú vhodným spôsobom obmedzené [fra].

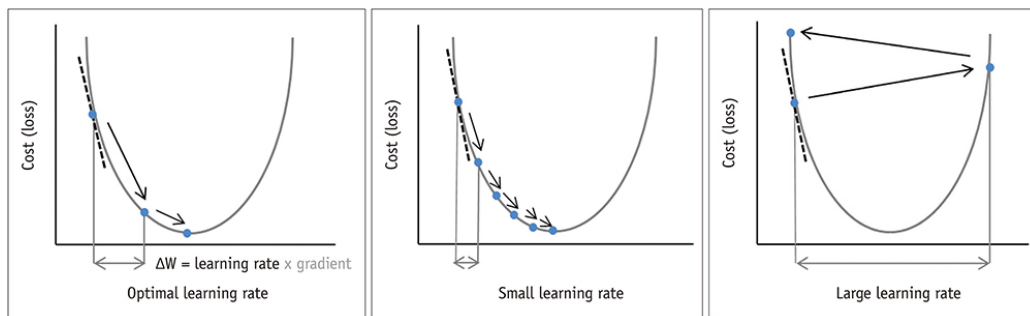
- Plne prepojená vrstva - prijíma vstupy z predchádzajúcej vrstvy a počíta skóre pre jednotlivé triedy a na výstupe vracia jednorozmerné pole veľkosti rovnajúcej sa počtu tried, neuróny tejto vrstvy sú prepojené tak ako v tradičných sieťach s každým číslom predošlého objemu [fra].

3.3 Hyperparametre

Hyperparametre neurónovej siete sú premenné, pomocou ktorých sa určuje štruktúra siete a ako je sieť trénovaná. Nastavujú sa vždy pred trénovaním siete.

Miera učenia

Miera učenia (learning rate) neurónovej siete určuje ako veľmi sa má model meniť v reakcii na odhadovanú chybu každej aktualizácie váh modelu. Miera učenia je jedným z najdôležitejších parametrov pri konfigurovaní neurónovej siete. Určenie ideálnej miery učenia je náročný proces, keďže pri veľmi nízkej hodnote to môže viesť k predĺženiu času učenia a naopak príliš vysoká hodnota môže viesť k nestabilnému učeniu [hyp].



Obr. 3.6: Ukážka rôznej miery učenia [DS20]

Počet epôch

Počet epôch určuje koľko krát algoritmus pri učení využije sadu trénovacích vzoriek. Pri každej iteračnej epoche sa neurónová sieť naučí niečo nové. Počet epoch sa zvyšuje, pokiaľ sa presnosť validácie nezačne znižovať aj keď sa pri tom znižuje presnosť tréningu.

Veľkosť dávky

Veľkosť dávky (batch size) určuje počet vzoriek, s ktorými neurónová sieť pracuje predtým ako upraví nastavenie parametrov. Po určitom počte vzoriek sa vždy výsledky porovnajú s pravdivým výsledkom a potom sa upraví parametre a prepočíta sa chyba.

3.4 Dataset

Pre tréning neurónovej siete je dôležité mať k dispozícii dostatočné množstvo dát.

Rozdelenie datasetu

Bežne sú dáta v datasete rozdelené do troch množín: trénovacia, testovacia a odporúča sa aj validačná množina. Štandardné rozdelenie dát do množín býva v pomere 60/20/20.

Vrstvy konvolučných neurónových sietí:

- Trénovacia množina,
- Validačná množina,
- Testovacia množina.

3.5 Detekcia a klasifikácia objektov

Patria sem algoritmy, ktoré využívajú strojové alebo hĺbkové učenie s cieľom získať čo najlepšie a zmysluplné výsledky. Princípom týchto metód je čo najviac sa priblížiť schopnosti nájsť zaujímavý objekt a rozpoznať ho ako to robí človek.

3.6 Detekcia objektov

Metódy detekcie objektov vo všeobecnosti patria do metód založených na strojovom učení alebo do metód založených na hlbokom učení. V prípade metód založených na strojovom učení je potreba najskôr určiť vlastnosti a následne pomocou rôznych techník vykonať klasifikáciu (napríklad metódou podporných vektorov (SVM)). Metódy založené na hlbokom učení dokážu detegovať rôzne objekty bez určenia vlastností a väčšinou sú založené na konvolučných neurónových sieťach.

V nasledujúcej časti sa budeme venovať najznámejším a najpoužívanejším metódam založeným na hlbokom učení [SSD20]:

- R-CNN,
- Fast R-CNN,
- Faster R-CNN,
- SSD,
- YOLO.

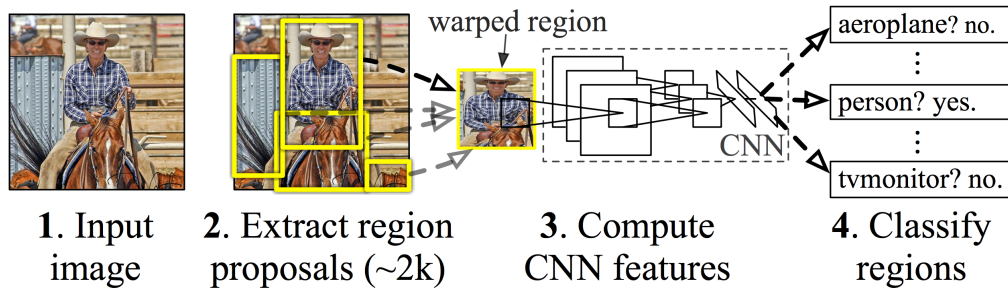
3.6.1 Rodina R-CNN (Region-based Convolutional Neural Networks)

Do tejto rodiny spadajú metóda R-CNN, ktorá bola základom pre jej vylepšenia Fast R-CNN a Faster R-CNN.

R-CNN

Princíp fungovania metódy R-CNN spočíva v tom, že najskôr sa pomocou selektívneho vyhľadávania vygenerujú navrhované regióny, ktoré by mohli obsahovať objekt, potom sa daným regiónom deformuje tvar na štvorcový a to je vstupom pre konvolučnú neurónovú sieť. Konvolučná sieť extrahuje príznaky a pomocou metódy podporných vektorov (SVM) sú klasifikované do tried [GDDM14].

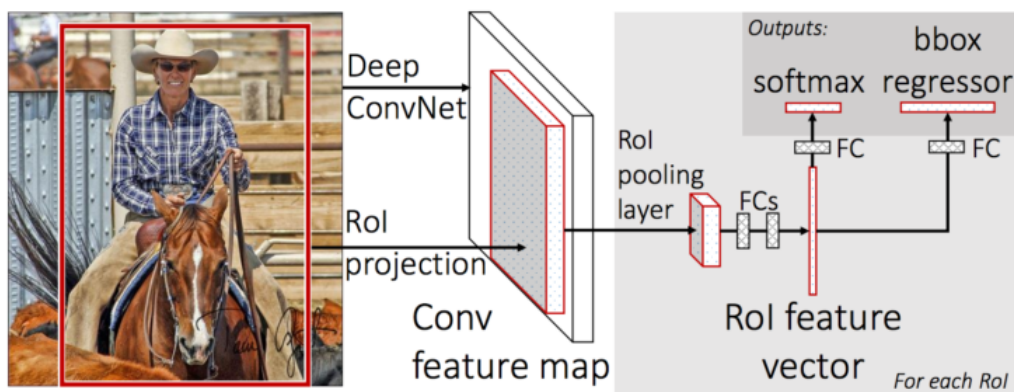
Hlavnou nevýhodou tejto metódy je selektívne vyhľadávanie, ktoré je pri veľkom počte navrhovaných regiónov, ktoré sa prekrývajú neefektívne.



Obr. 3.7: Príklad RCNN architektúry [GDDM14]

Fast R-CNN

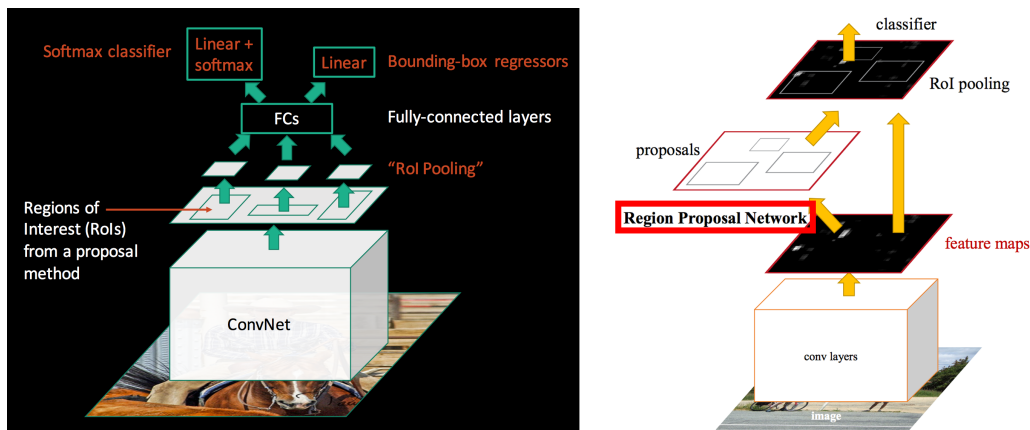
Pomocou tejto metódy sa podarilo odstrániť niektoré nedostatky pôvodnej R-CNN metódy, ako napríklad prekryvanie navrhovaných regiónov. Vstupný obrázok prichádza priamo do konvolučnej neurónovej siete, ktorá generuje konvolučná mapa vlastností. Z tejto mapy sa ďalej pomocou selektívneho vyhľadávania nájdu návrhy regiónov, ktoré sa deformujú do štvorcov. Následne sa návrhy pomocou RoI (Region of Interest) vrstvy spoja do vektora s pevnou veľkosťou, ktorý je vstupom pre plne prepojenú vrstvu, ktorá za pomoci dvoch výstupných vrstiev vytvorí odhad pravdepodobnosti a štyri súradnice pre ohraničujúci box pre každú triedu [Gir15].



Obr. 3.8: Príklad Fast RCNN architektúry [Gir15]

Faster R-CNN

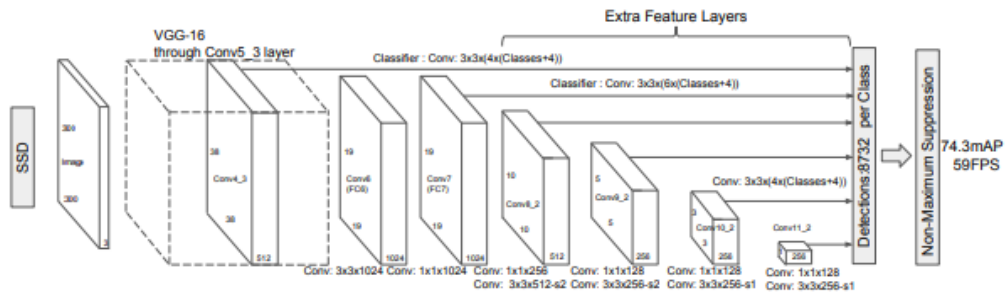
Selektívne vyhľadávanie je časovo náročný proces a preto sa ho autori Faster R-CNN rozhodli v tejto metóde nepoužiť. Miesto neho sa na návrh regiónov používa samostatná sieť. Systém sa teda skladá z dvoch častí. Prvou časťou je hlboká konvolučná sieť, ktorá navrhuje regióny a druhá časť je R-CNN detektor, ktorý ich ďalej používa. Celý systém je jednotná, plne prepojená sieť [RHGS17]. Vďaka odstráneniu selektívneho vyhľadávania sa celý proces značne zrýchlil a táto metóda sa dala využiť aj pri detekcii objektov v reálnom čase.



Obr. 3.9: Príklad Faster RCNN architektúry [RHGS17]

3.6.2 SSD (Single Shot Detector)

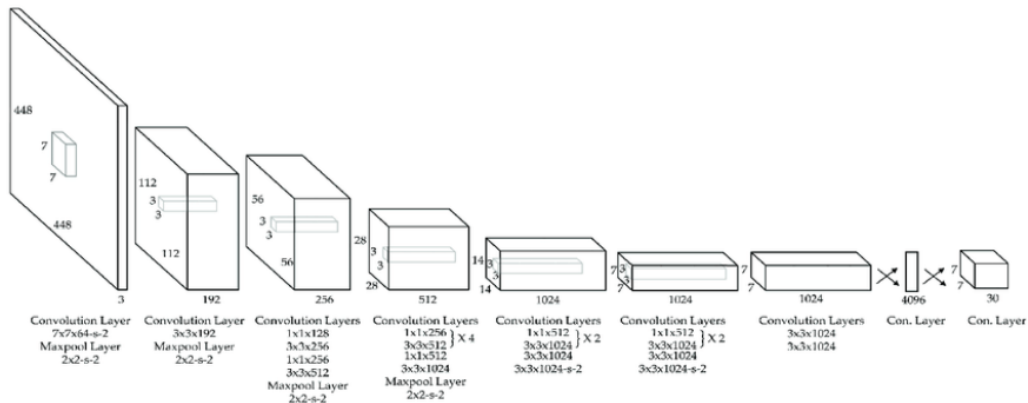
Tento detektor bol vytvorený pre detekciu objektov v reálnom čase. SSD využíva skrátený model konvolučnej neurónovej siete VGG-16, ktorá sa aplikuje na obrázok iba raz a vypočíta mapu vlastností. Potom použije 3x3 konvolučné filtre pre každú bunku na vytvorenie predikcii a pomocou týchto filtrov sa vypočíta skóre polohy a tried [LAE⁺16].

Obr. 3.10: Príklad SSD architektúry [LAE⁺16]

3.6.3 YOLO (You Only Look Once)

YOLO je tvorený jednou neurónovou sieťou, ktorá spracováva všetky objekty naraz, čím dosahuje veľmi dobrú rýchlosť a zachovanie presnosti. Najväčšou výhodou tohto modelu je obrovská rýchlosť, vďaka ktorej je veľmi populárny a používaný [RDGF16].

Dnes máme k dispozícii viacero verzii tejto architektúry ako napríklad YOLOv2, YOLOv3, YOLOv5.



Obr. 3.11: Príklad YOLO architektúry [RDGF16]

Kapitola 4

Návrh riešenia problematiky

Cieľom našej práce je detegovať reklamné plochy popri cestách a detegovať ich významnosť. Postup riešenia tejto problematiky si môžeme logicky rozdeliť na dva veľké celky:

- Detekcia reklamnej plochy.
- Detekcia významnosti reklamnej plochy.

4.1 Detekcia reklamnej plochy

V tejto časti budeme pracovať s jednotlivými snímkami získanými z videa.

Dataset

Prvým krokom, ktorý budeme musieť riešiť je výber vhodného datasetu. Keďže pri riešení budeme používať neurónové siete, na tréning bude potreba veľkého množstva obrázkov. V našom prípade máme dve možné riešenia. Prvým je tvorba vlastného datasetu, čo je veľmi náročný a zdĺhavý proces, pretože je potreba získať veľké množstvo snímok z ciest alebo miest a následne

tieto obrázky aj ručne anotovať. Druhou možnosťou je použitie existujúceho datasetu. Pri výbere správneho datasetu musíme byť pozorný, aby obsahoval dostatočné množstvo snímok s objektami, ktoré budeme detekovať - teda reklamné plochy. Jedným z takýchto datasetov by mohol byť Mapillary Vistas Dataset [NOBK17]. Je to dataset vytvorený pre detekciu rôznych objektov. Obsahuje snímky s vysokým rozlíšením a sú anotované.

Model

Ďalším krokom bude výber modelu. Keďže detekcia reklamných plôch bude prebiehať v reálnom čase, dôležitým krokom bude vybrať rýchly, ale aj presný model. Pre riešenie tejto problematiky by mohol byť vhodný model YOLO, ktorý momentálne veľmi populárny, pretože dosahuje veľmi dobré výsledky, čo sa týka rýchlosti za zachovania presnosti detekcie. Ideálne bude nájsť nejaký predtrénovaný model, ktorý je určený na detekciu objektov, čo nám výrazne skráti čas tréningu a potrebu množstva obrázkov.

4.2 Detekcia významnosti reklamnej plochy

V tejto časti budeme pracovať už len so získanými oblasťami, ktoré sme získali zo snímok z videa. Najskôr bude potreba určiť vlastnosti, na základe ktorých sa bude vyhodnocovať významnosť. Medzi tieto vlastnosti by mohla patriť napríklad farebnosť reklamných plôch. Významnosť reklamných plôch budeme overovať pomocou zariadenia eyetracker, ktorý máme k dispozícii v škole.

Kapitola 5

Riešenie problematiky a implementácia

Pred samotným začiatkom tréningu modelu, ktorý chceme naučiť detegovať reklamné plochy, budeme musieť urobiť niekoľko krokov. Ako prvý krok budeme musieť pripraviť dataset a potrebné súbory pre tréning, ktoré obsahujú údaje o anotovaných objektoch na jednotlivých obrázkoch.

Ďalším krokom je trénovanie modelu, pre ktorý sme si vybrali model s predtrénovanými váhami, ktorý dotrénujeme pomocou nových obrázkov. Po skončení trénovania budeme mať hneď k dispozícii výsledky.

5.1 YOLOv5

Na riešenie problematiky sme si vybrali jeden z najpopulárnejších modelov na detekciu objektov v reálnom čase. Model YOLO patrí medzi presné a veľmi rýchle modely. V našom riešení použijeme najaktuálnejšiu dostupnú verziu YOLOv5, ktorá je voľne dostupná na GitHubu [yol], kde máme k dispozícii na stiahnutie aj predtrénované váhy. K dispozícii je 5 rôznych váh

(YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x). Na GitHube sa k týmto váham nachádza aj tabuľka, kde sú vykonané rôzne metriky (napr. presnosť, mAP, rýchlosť) a užívatelia si môžu vybrať podľa potreby.

5.2 Použité nástroje

5.2.1 Pytorch

Pytorch je open-source knižnica založená na knižnici Torch a určená primárne na účely strojového učenia (machine learning) a hĺbkového učenia (deep learning). Táto knižnica sa najčastejšie používa v kombinácii s jazykom Python, avšak je možné ju použiť aj s jazykom C++. Je vyvíjaná hlavne výskumným tímom Facebooku pre umelú inteligenciu (Facebook's AI Research lab) [pyt].

Je jednoduchý na naučenie a používanie. Obsahuje rozsiahlu dokumentáciu a rôzne tutoriály. Poskytuje možnosť rýchleho počítania tzv. tensorov (forma výstupu v knižnici Pytorch podobná matici) použitím GPU.

5.2.2 Weights & Biases

WandB je nástroj určený na sledovanie tréningu modelov. Slúži na presnejšie nastavovanie hyperparametrov, zobrazenie metrík a predpovedí [wan].

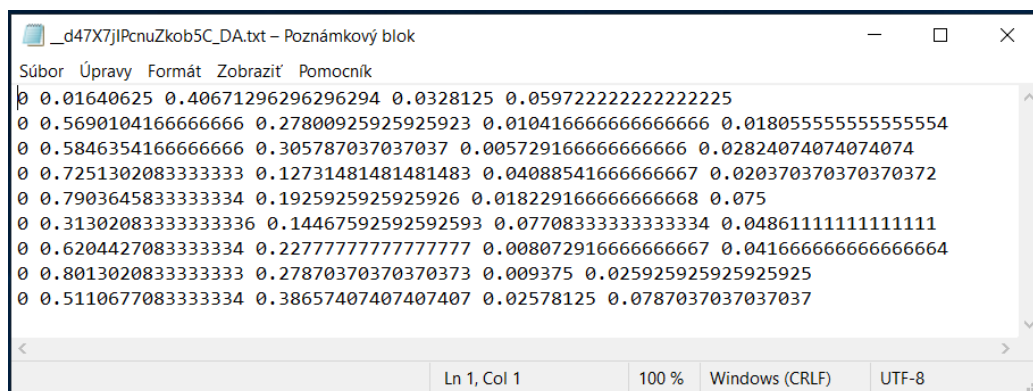
5.3 Dataset

Pri trénovaní sme použili Mapillary Vistas dataset. Obsahuje 25000 obrázkov s vysokým aj nižším rozlíšením. Anotácie objektov so všetkými informáciami o objektoch sa nachádzajú v jednom .json súbore pre trénovaciu a validačnú množinu. Obrázky v datasete sú rozložené do troch množín nasledovne:

- Trénovacia množina - 18000 obrázkov,
- Validačná množina - 2000 obrázkov,
- Testovacia množina - 5000 obrázkov.

5.3.1 Príprava datasetu a súborov pre tréning

Pred začatím tréningu sme museli pripraviť potrebné súbory datasetu. Model YOLOv5 si vyžaduje textový súbor s anotáciami objektov s rovnakým názvom ako prislúchajúci obrázok. V jednom riadku tohto textového súboru sa nachádzajú informácie o jednom detekovanom objekte na obrázku s tým, že nájdených objektov môže byť viac, takže súbor môže obsahovať viacero riadkov, ale aj žiaden. Informácie o objekte sú formátované v súbore nasledovne: (id, S_x , S_y , w , h), kde id je označenie triedy, do ktorej objekt patrí, S_x a S_y sú súradnice stredu obdĺžnika opísaného okolo objektu, w je šírka a h je výška obrázku. S_x , S_y , w , h sú normalizované na veľkosť obrázku. V našom prípade bolo id vždy 0, keďže sme detegovali objekty iba jednej triedy - reklamné plochy.



Obr. 5.1: Príklad textového súboru s anotáciami objektov

5.4 Tréning

Pri trénovaní modelov sme využili už predtrénovaný model YOLOv5, resp. váhy z predtrénovaného modelu, čo nám urýchlilo trénovanie. TODO.

Kapitola 6

Vyhodnotenie a výsledky

TBD

Literatúra

- [DS20] Chung Joo Won Do Synho, Song Kyoung Doo. Basics of deep learning: A radiologist’s guide to understanding published radiology articles on deep learning. *kjr*, 21(1):33–41, 2020.
- [fra] Framework torch. [https://www.root.cz/clanky/framework-torch-vyuziti-konvolucnich-siti-pro-rozpoznávání-a-klasifikaci/#k0](https://www.root.cz/clanky/framework-torch-vyuziti-konvolucnich-siti-pro-rozpoznávání-a-klasifikaci/). Navštívené: 2021-12-8.
- [GBD⁺18] Reagan Galvez, Argel Bandala, Elmer Dadios, Ryan Vicerra, and Jose Martin Maningo. Object detection using convolutional neural networks. pages 2023–2027, 10 2018.
- [GDDM14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.
- [Gir15] Ross Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.
- [Hay08] Simon Haykin. *Neural Networks and Learning Machines*. Prentice Hall, 2008. Third Edition.

[HDN⁺18] Murhaf Hossari, Soumyabrata Dev, Matthew Nicholson, Killian McCabe, Atul Nautiyal, Clare Conran, Jian Tang, Wei Xu, and François Pitié. Adnet: A deep network for detecting adverts, 2018.

[hyp] What are hyperparameters? <https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a>
Navštívené: 2021-12-8.

[IKN98] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[KZS19] Caglar Koylu, Chang Zhao, and Wei Shao. Deep neural networks and kernel density estimation for detecting human activity patterns from geo-tagged images: A case study of birdwatching on flickr. *ISPRS International Journal of Geo-Information*, 8:45, 01 2019.

[LAE⁺16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *Lecture Notes in Computer Science*, page 21–37, 2016.

[NOBK17] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kotschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5000–5009, 2017.

[pyt] Pytorch. <https://pytorch.org/>.

- [RDGF16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.
- [RDS⁺19] Romi Fadillah Rahmat, Dennis Dennis, Opim Salim Sitompul, Sarah Purnamawati, and Rahmat Budiarto. Advertisement billboard detection and geotagging system with inductive transfer learning in deep convolutional neural network. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 2019.
- [RHGS17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.
- [SSD20] F. Sultana, A. Sufian, and P. Dutta. *A Review of Object Detection Models Based on Convolutional Neural Network*, pages 1–16. Springer Singapore, Singapore, 2020.
- [TKDK18] Akash Tripathi, T.V. Kumar, Tarun Dhansetty, and J. Kumar. Real time object detection using cnn. *International Journal of Engineering and Technology(UAE)*, 7:33–36, 04 2018.
- [wan] Weights and biases. <https://wandb.ai/site>.
- [yol] YOLOv5. <https://github.com/ultralytics/yolov5>.
- [18] Aleksa Ćorović, Velibor Ilić, Siniša Đurić, Mališa Marijan, and Bogdan Pavković. The real-time detection of traffic participants using yolo algorithm. In *2018 26th Telecommunications Forum (TELFOR)*, pages 1–4, 2018.

- [B⁺11] Elena Šikudová, Zuzana Černeková, Wanda Benešová, Zuzana Haladová, and Júlia Kučerová. *Počítačové videnie Detekcia a rozpoznávanie objektov*. Vydavateľstvo Wikina, 2011.

Zoznam obrázkov

2.1	Architektúra siete AlexNet	4
2.2	Architektúra siete ADNet	5
2.3	Architektúra siete YOLO	6
2.4	Výsledok detekcie účastníkov cestnej premávky	7
3.1	Eyetracker	9
3.2	Ittiho model	10
3.3	Schéma biologického neurónu	11
3.4	Schéma umelého neurónu	12
3.5	Ukážka konvolučnej neurónovej siete	14
3.6	Príklad miery učenia	16
3.7	Príklad RCNN architektúry	19
3.8	Príklad Fast RCNN architektúry	19
3.9	Príklad Faster RCNN architektúry	20
3.10	Príklad SSD architektúry	21
3.11	Príklad YOLO architektúry	21
5.1	Príklad textového súboru s anotáciami objektov	26