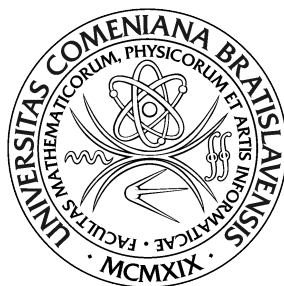


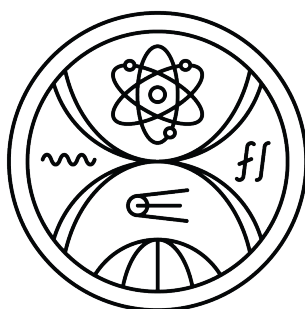
UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



DETEKCIA VIZUÁLNEHO SMOGU PRI CESTÁCH

Diplomová práca

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.



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Michal Zrubec
Študijný program: aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Detekcia vizuálneho smogu pri cestách
Detection of visual smog near roads

Anotácia: Naštudovať metódy spracovania obrazu, detekcie významných oblastí a neurónové siete. Preskúmať možnosť využitia neurónových sietí na detekciu reklamných plôch popri cestách. Detekovať významnosť reklám na základe saliency a následne verifikovať pomocou eyetrackera. Navrhnutú metódu otestovať a vyhodnotiť výsledky.

Cieľ: Naštudovať metódy spracovania obrazu, detekcie významných oblastí a neurónové siete. Preskúmať možnosť využitia neurónových sietí na detekciu reklamných plôch popri cestách. Detekovať významnosť reklám na základe saliency a následne verifikovať pomocou eyetrackera. Navrhnutú metódu otestovať a vyhodnotiť výsledky.

Vedúci: RNDr. Zuzana Černeková, PhD.
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.
Dátum zadania: 08.10.2017

Dátum schválenia: 23.09.2019

prof. RNDr. Roman Ďurikovič, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

PodĎakovanie: Ďakujem mojej školiteľke RNDr. Zuzane Černekovej, PhD. za všetok poskytnutý čas, konzultácie, poskytnuté materiály, cenné rady a pomoc pri riešení tejto práce. Táto diplomová práca vznikla aj vďaka podpore v rámci Operačného programu Integrovaná infraštruktúra pre projekt: Advancing University Capacity and Competence in Research, Development a Innovation (ACCORD), spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

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 úvodnej teoretickej časti práce sú popísané reklamné plochy a podobné riešenia. Dôraz je kladený na popis neurónových sietí a významných oblastí. Práca obsahuje popis návrhu riešenia problému a v nadväznosti na návrh je popísaná samotná implementácia konkrétnej metódy na detekciu reklamných plôch a významnosti reklamných plôch. V poslednej časti práce sa nachádzajú vyhodnotenia metód detekcií reklamných plôch a detekcií významnosti reklamných plôch.

Kľúčové slová: detekcia, neurónové siete, významnosť, reklamné plochy

Abstract

The aim of this work is to examine the possibility of using neural networks to detect advertising space along the roads, detect the significance of ads based on saliency and then verify with an eyetracker. The introductory theoretical part of the thesis is focused on the description of advertising space and similar works. Emphasis is placed on the description of the neural networks and important areas in more detail. The thesis contains a description of the proposed solution to the problem and, following the proposal, the implementation of a specific method for the detection of advertising space and the importance of advertising space is described. In the last part of the work there are evaluations of methods for detection of advertising space and detection of saliency of advertising space.

Keywords: detection, neural networks, saliency, advertising space

Obsah

Úvod	1
1 Prehľad problematiky	2
1.1 Detekcia objektov	2
1.2 Motivácia detekcie reklamných plôch	2
1.3 Podobné práce	3
1.3.1 Detekcia billboardov	3
1.3.2 Detekcia reklám	3
1.3.3 Detekcia účastníkov cestnej premávky	4
1.3.4 Výskum modelov významných oblastí malieb	5
2 Teoretická časť	7
2.1 Neurónové siete a umelá inteligencia	7
2.1.1 Biologický neurón	7
2.1.2 Umelý neurón	8
2.1.3 Umelé neurónové siete	8
2.1.4 Konvolučné neurónové siete	9
2.2 Aktivačná funkcia	10
2.3 Loss function (Stratová funkcia)	11
2.4 Hyperparametre	11
2.5 Detekcia objektov	12
2.5.1 Rodina R-CNN (Region-based Convolutional Neural Networks)	13
2.5.2 SSD (Single Shot Detector)	15
2.5.3 YOLO (You Only Look Once)	15
2.6 Datasetsy	16

2.7	Detekcia významných oblastí v obraze	17
2.7.1	Metódy na detekciu významných oblastí	19
2.8	Datasety	21
3	Návrh riešenia problematiky	22
3.1	Detekcia reklamnej plochy	22
3.2	Detekcia významnosti reklamných plôch	23
4	Riešenie problematiky a implementácia	25
4.1	Pytorch	25
4.2	Eyetracker Tobii Pro Glasses 3	26
4.3	Dataset	26
4.3.1	Mappilary Vistas Dataset	26
4.3.2	Vlastný dataset	27
4.3.3	Príprava datasetov a súborov	28
4.4	Transfer learning	30
4.4.1	Trénovanie modelov	32
4.5	Významnosť reklamnej plochy	33
4.5.1	OpenCV Saliency API	34
4.5.2	UniSal	34
4.5.3	Detekcia významnosti reklamnej plochy	35
5	Vyhodnotenie a výsledky	37
5.1	Detekcia reklamnej plochy	37
5.1.1	Použité metriky	37
5.1.2	Experimenty	40
5.1.3	Porovnanie výsledkov	41
5.2	Detekcia významnosti	43
5.2.1	Použité metriky	43
5.2.2	Porovnanie metód na detekciu významnosti	45
5.2.3	Detekcia významnosti reklamnej plochy	45
5.3	Zhrnutie výsledkov	46
	Záver	49

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 sa viaže 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.

Detekcia významných oblastí patrí v poslednej dobe medzi veľmi žiadané oblasti výskumu počítačového videnia. Simulovanie ľudskej pozornosti môže mať rôzne využitia v priemysle, ale aj v umení. Jedným z využití detekcie významných oblastí je detekcia objektov na základe najvýznamnejších oblastí v obraze.

V práci si postupne predstavíme prehľad problematiky so stručne popísanými podobnými riešeniami. V teoretickej časti sa zameriame na neurónové siete, detekcie objektov a významností a ďalšie potrebné teoretické znalosti. Potom si vytvoríme návrh riešenia, ktoré implementujeme v ďalšej kapitole a na záver toto riešenie otestujeme a vyhodnotíme.

Kapitola 1

Prehľad problematiky

1.1 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.

1.2 Motivácia detekcie reklamných plôch

Motiváciou pre túto prácu 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. Ďalšou motiváciou by mohlo byť zistenie informácií o tom, ako veľmi reklamné plochy pri cestách ovplyvňujú správanie vodičov.

1.3 Podobné práce

1.3.1 Detekcia billboardov

Cieľom práce autorov bolo vytvorenie systému, ktorý má za úlohu detekciu reklamných billboardov a označovanie ich polohy pomocou hlbkej konvolučnej neurónovej siete [1].

Pri riešení problému sa autori rozhodli použiť predtrénovaný model, ktorý používal architektúru AlexNet hlbkej 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. 1.1: Ukážka použitej architektúry [1].

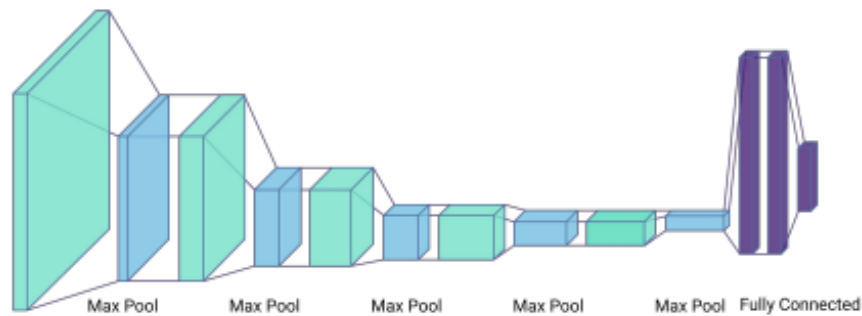
Pri trénoch siete autori vykonali dva experimenty. Pri prvom sa vykonalo 4000 krokov trénochovania s presnosťou 89,3% a pri druhom sa vykonalo 8000 krokov trénochovania s presnosťou 92,7%. Pri testovaní systému dosiahli presnosť 72%, kde problém robili hlavne billboardy, ktoré sa nachádzali v obrázkoch ďaleko s presnosťou detekcie a určenia polohy iba 28,1%.

1.3.2 Detekcia reklám

Cieľom tejto práce bolo vytvoriť systém s využitím neurónovej siete, ktorý zisťuje, či sa na danej snímke videa nachádza nejaká reklamná plocha [2].

Na riešenie tohto problému autori vytvorili sieť ADNet (advert-detection model), ktorá bola inšpirovaná VGG19 architektúrou. Model tejto siete mal veľmi malé konvolučné filtre (3×3) 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.



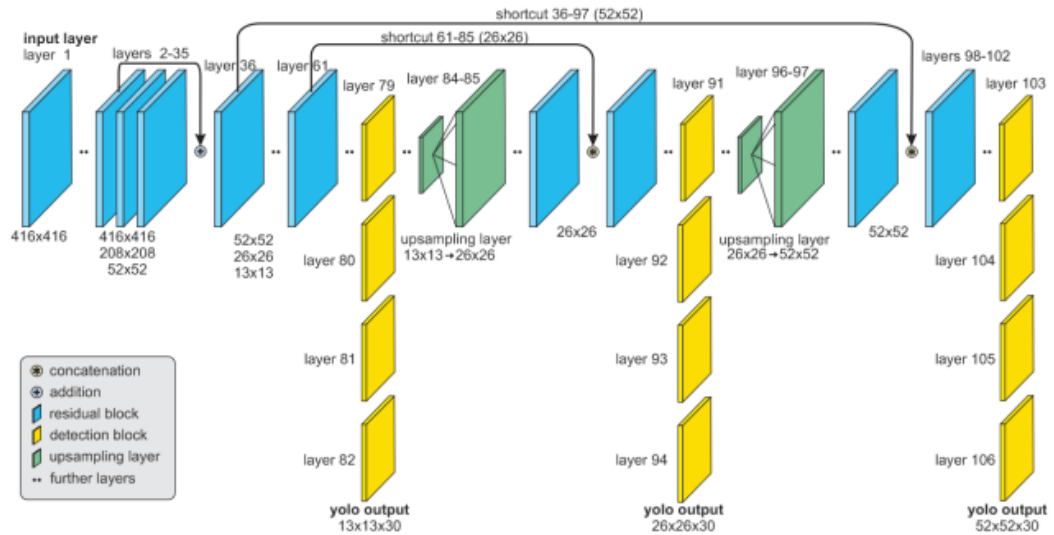
Obr. 1.2: Ukážka použitej architektúry [2].

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

1.3.3 Detekcia účastníkov cestnej premávky

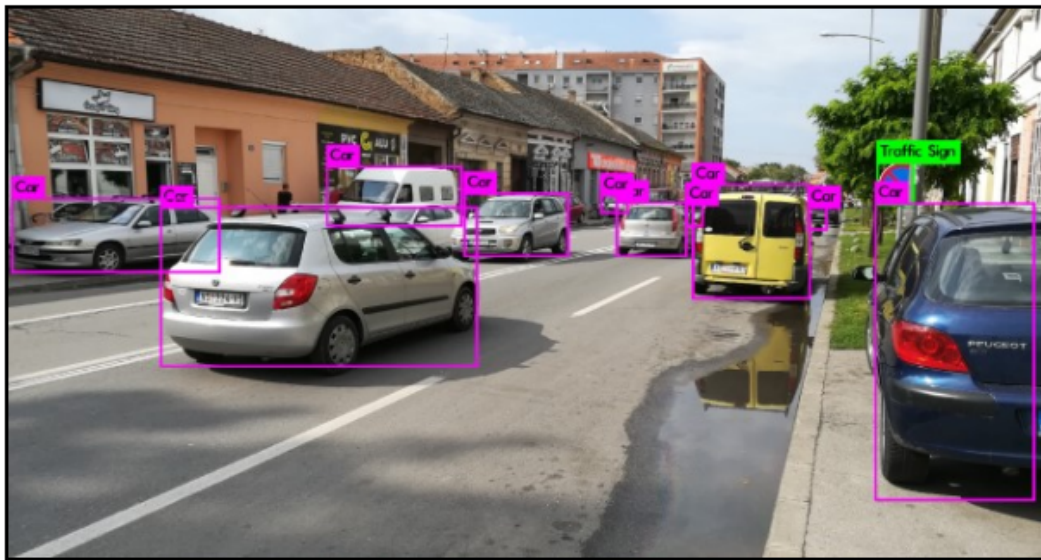
Autori sa v tomto článku zaoberali detekciou účastníkov cestnej premávky pomocou neuronových sietí. Zamerali sa pritom na klasifikáciu áut, kamiónov, chodcov, dopravných značiek a semaforov [3].

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



Obr. 1.3: Ukážka použitej architektúry [3].

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



Obr. 1.4: Výsledok detekcie účastníkov cestnej premávky [3].

1.3.4 Výskum modelov významných oblastí malieb

Autorka sa v práci venovala analýze saliency modelov, za účelom nájsť taký model, ktorý dokáže najlepšie zachytiť pohľad umelcov na umelecké dielo *Posledná večera* [4].

V práci boli implementované tri modely, ktoré boli otestované na dátach od 35 respondentov:

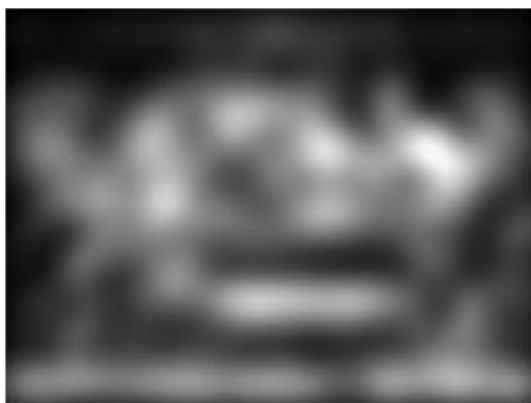
- A Model of Saliency-based Visual Attention for Rapid Scene Analysis,

- Random Center Surround Saliency (RCSS),
- Saliency Attentive Models (SAM-VGG and SAM-ResNet).

Modely boli vyhodnotené pomocou troch rôznych metrík, kde najlepšie výsledky boli získané modelmi, ktoré využívajú prístup hlbokého učenia - SAM-ResNet a SAM-VGG model.



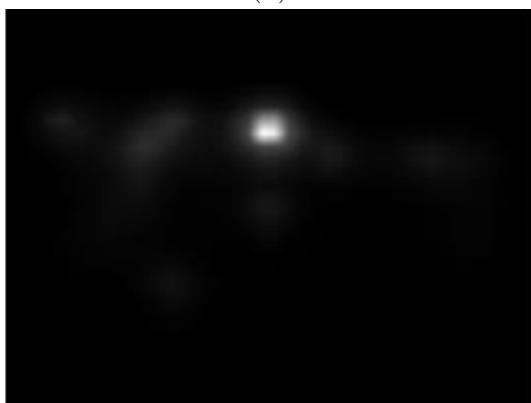
(a)



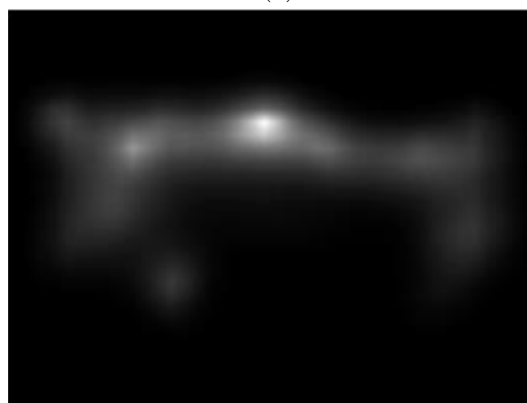
(b)



(c)



(d)



(e)

Obr. 1.5: Fixácie očí na diele a príslušné saliency mapy z rôznych modelov. a) - Fixácie očí na pôvodný obrázok od Juanesa. b) - Saliency mapa vytvorená IttiKoch modelom. c) - Saliency mapa vytvorená RCSS modelom. d) - Saliency mapa vytvorená SAM-VGG modelom. e) - Saliency mapa vytvorená SAM-ResNet modelom [4].

Kapitola 2

Teoretická časť

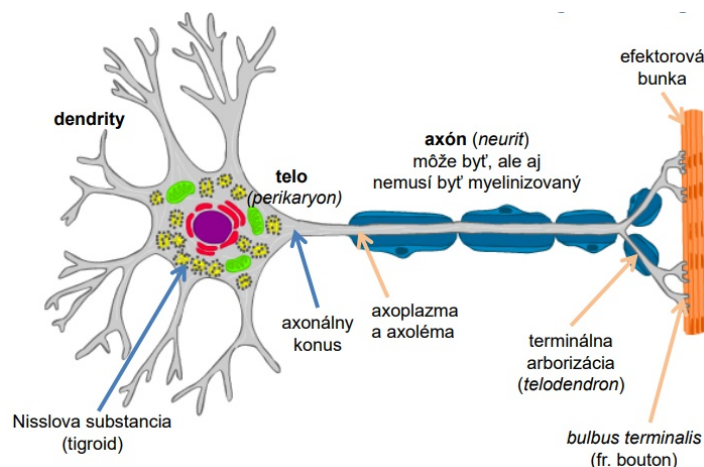
2.1 Neurónové siete a umelá inteligencia

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

2.1.1 Biologický neurón

Biologický neurón je nezávislá atomická a funkčná jednotka, ktorej úlohou je získavanie, prenos, spracovanie a ukladanie informácií [5]. Základnými časťami neurónu sú:

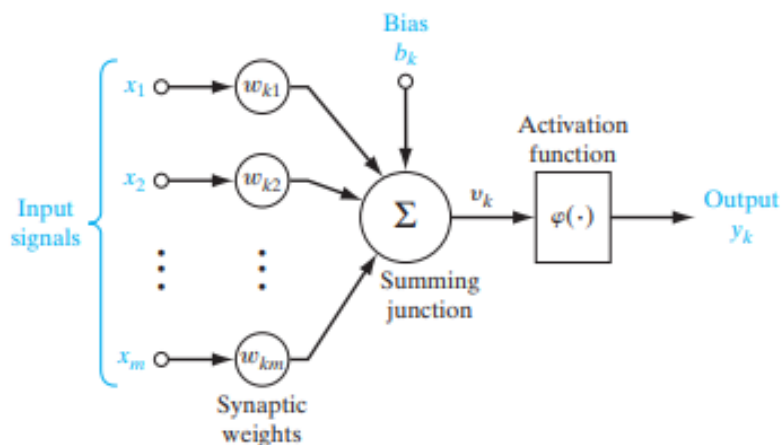
- Neurocyt alebo perikaryón - telo neurónu,
- Dendrit - prijíma vstupnú informáciu z okolia od ostatných neurónov,
- Neurit alebo axón - vedie informáciu od tela neurónu iným neurónom do okolia.



Obr. 2.1: Schéma biologického neurónu.

2.1.2 Umelý neurón

Základnou funkciou umelého neurónu je zber elementárnych informácií, ich vyhodnotenie a generovanie zodpovedného výstupu. Neurón môže mať ľubovoľný 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 [6].



Obr. 2.2: Schéma umelého neurónu [6].

2.1.3 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 [6].

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 zároveň vstupom do neurónovej siete. Prenosová funkcia reprezentuje lineárnu funkciu, takže neuróny odosiajú bez zmeny vstupný signál ako výstupný. Výstupná vrstva definuje výstup neurónovej siete [6].

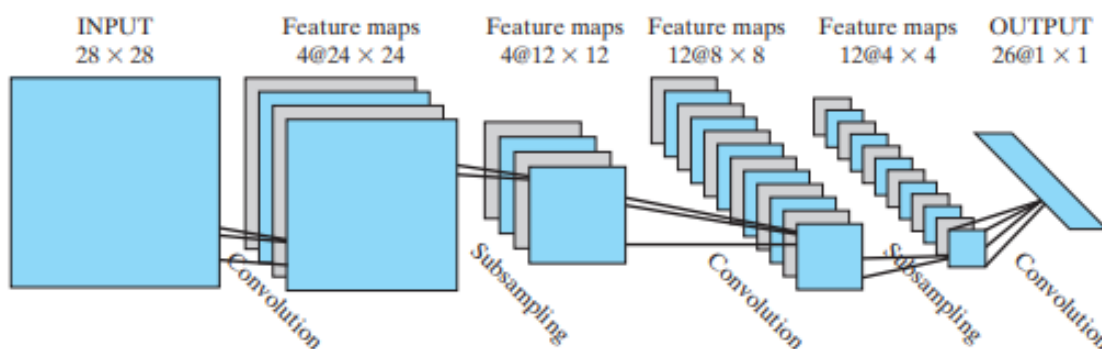
Neurónová sieť sa dá charakterizovať na základe niekoľkých vlastností. Medzi najhlavnejšie patrí topológia siete, spôsob učenia, atď.

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. Pri umelých neurónových sieťach 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.

2.1.4 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. 2.3: Príklad konvolučnej neurónovej siete [6].

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).

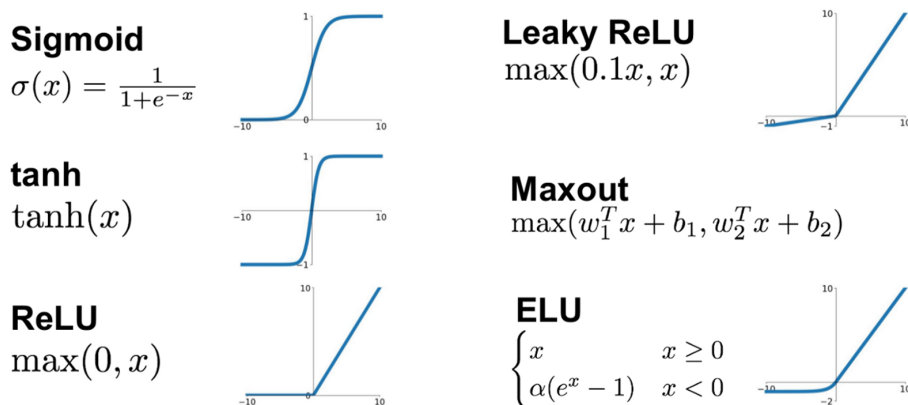
- Konvolučná vrstva - jej úlohou je zistiť lokálne informácie rastrového obrázku. Jedna konvolučná vrstva obsahuje niekoľko filtrov. Dôvodom je, že každý filter sa zameriava na inú vlastnosť obrázka a ostatné ignoruje. Napríklad filtre v prvej vrstve neurónovej siete môžu detegovať hrany a to každý v inom smere.
- Poolingová vrstva - na svojom vstupe akceptuje výstup z predchádzajúcej konvolyčnej 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 konvolyčná $\rightarrow$ poolingová $\rightarrow$ konvolyčná, dochádza k postupnému znižovaniu množstva dát, takže vstupy do poslednej časti siete sú vhodným spôsobom obmedzené [7].
- 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 [7].

2.2 Aktivačná funkcia

Aktivačná funkcia je funkcia umelého neurónu, ktorá na základe vstupných údajov neurónu vytvorí jeho výstup. Výber správnej aktivačnej funkcie môže mať veľký dopad na dĺžku učenia.

Medzi najznámejšie aktivačné funkcie patria:

- Sigmoid,
- $\tanh(x)$ - Hyperbolický tangens,
- ReLU - Rektifikovaná lineárna jednotka,
- Leaky ReLU - tečúce ReLU,
- ELU - Exponenciálne lineárne jednotky,
- Selu - Škálované exponenciálne lineárne jednotky,
- Maxout "neurón".



Obr. 2.4: Grafy a vzorce aktivačných funkcií [8].

2.3 Loss function (Stratová funkcia)

Stratová funkcia je jeden z najdôležitejších údajov pri tréňovaní. Model neurónovej siete pomocou stratovej funkcie vypočítava rozdiel medzi predikovaným a skutočným údajom.

Stratu na celej tréningovej množine môžeme vypočítať ako súčet strát na jednotlivých častiach tréningovej množiny:

$$Loss = \frac{1}{N} \sum_i L_i,$$

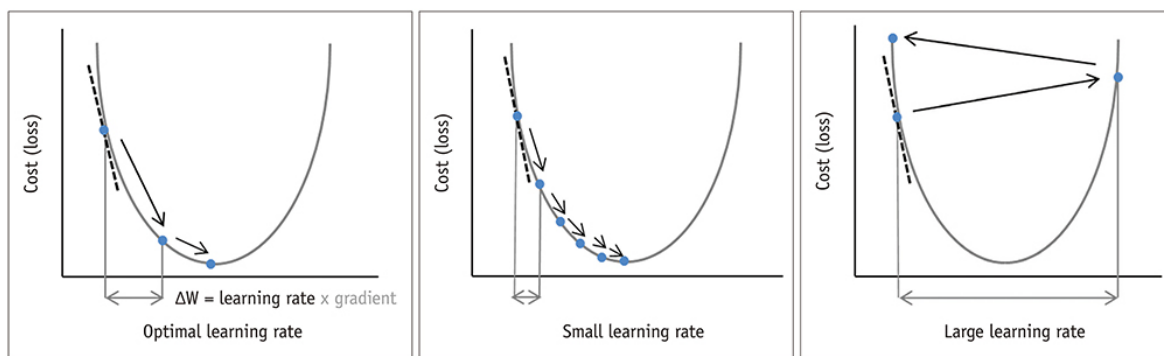
kde N je počet častí tréningovej množiny.

2.4 Hyperparametre

Hyperparametre neurónovej siete sú premenné, pomocou ktorých sa určuje štruktúra siete, ako aj to ako je sieť tréňovaná. Nastavujú sa vždy pred tréňovaním 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 [9].



Obr. 2.5: Ukážka rôznej miery učenia [10].

Počet epôch

Počet epôch určuje, koľkokrát algoritmus pri učení využije sadu trénovacích vzoriek. Pri každej iteračnej epoche sa neurónová sieť naučí niečo nové. Počet epôch sa zvyšuje, až 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 upravia parametre a prepočíta sa chyba.

2.5 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 [11]:

- R-CNN,
- Fast R-CNN,

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

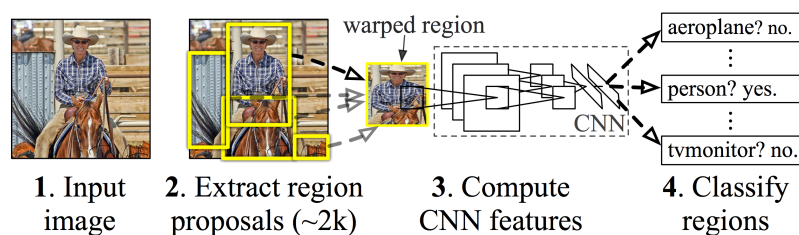
2.5.1 Rodina R-CNN (Region-based Convolutional Neural Networks)

Do tejto rodiny spadá 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 sa najskôr pomocou selektívneho vyhľadávania vygenerujú navrhované regióny, ktoré by mohli obsahovať objekt, potom sa daným regiónom deformuje tvar na štvorcový, ktorý je vstupom pre konvolučnú neurónovú sieť. Konvolučná sieť extrahuje príznaky a pomocou metódy podporných vektorov (SVM) sú dané regióny klasifikované do tried [12].

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

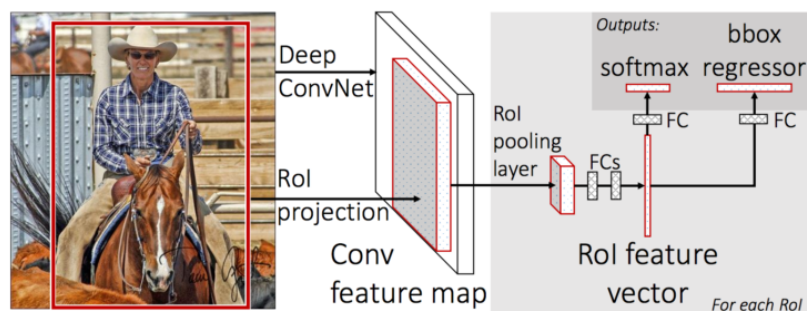


Obr. 2.6: RCNN architektúra [12].

Fast R-CNN

Pomocou tejto metódy sa podarilo odstrániť niektoré nedostatky pôvodnej R-CNN metódy, ako napríklad prekrývanie navrhovaných regiónov. Vstupný obrázok prichádza priamo do konvolučnej neurónovej siete, ktorá generuje konvolučnú mapu 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

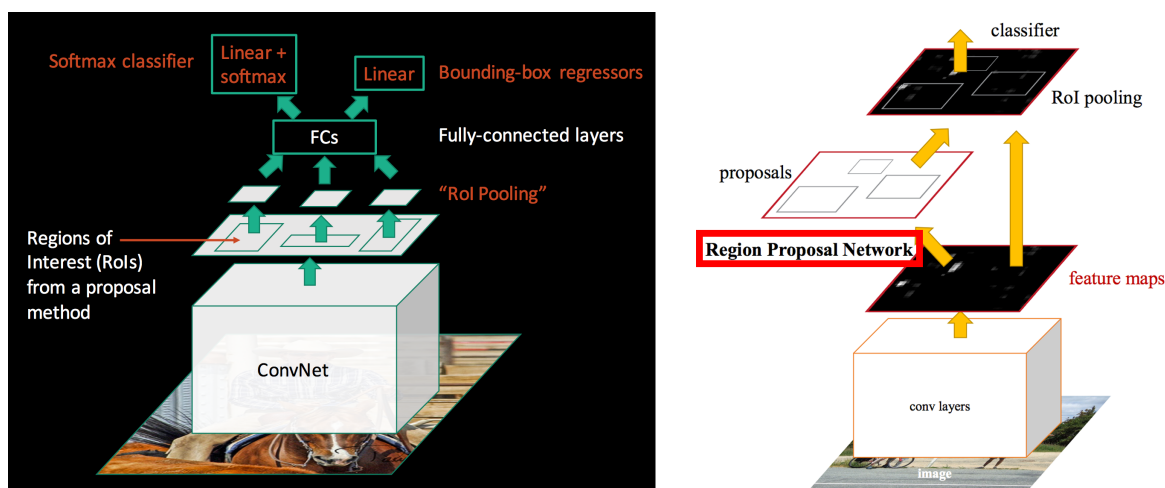
vrstvu, pričom za pomoci dvoch výstupných vrstiev vytvorí odhad pravdepodobnosti a štyri súradnice pre ohraňujúci box pre každú triedu [13].



Obr. 2.7: Fast RCNN architektúra [13].

Faster R-CNN

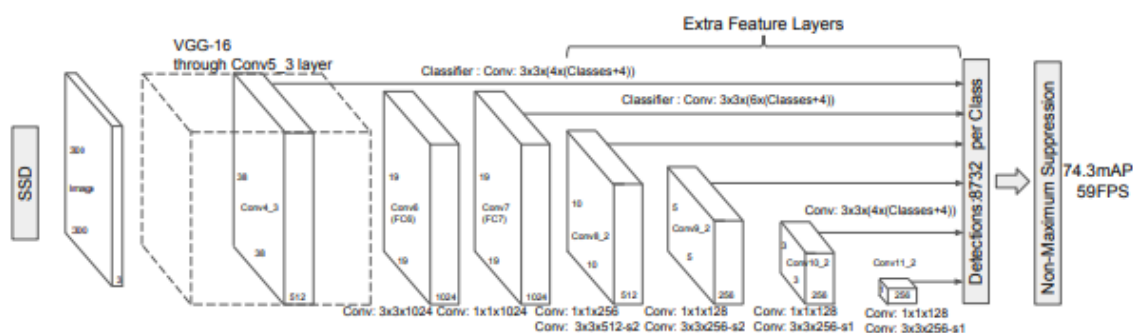
Selektívne vyhľadávanie je časovo náročný proces, a preto sa ho autori Faster R-CNN rozhodli v tejto metóde nepoužiť. Miesto neho sa na návrh regiónov používa samostatná sieť. Systém sa teda skladá z dvoch častí. Prvou časťou je hlboká konvolučná sieť, ktorá navrhuje regióny a druhá časť je R-CNN detektor, ktorý ich ďalej používa. Celý systém je jednotná, plne prepojená sieť [14]. Vďaka odstráneniu selektívneho vyhľadávania sa celý proces značne zrýchlil a táto metóda sa dala využiť aj pri detekcii objektov v reálnom čase.



Obr. 2.8: Faster RCNN architektúra [14].

2.5.2 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 3×3 konvolučné filtre pre každú bunku na vytvorenie predikcii a pomocou týchto filtrov sa vypočíta skóre polohy a tried [15].

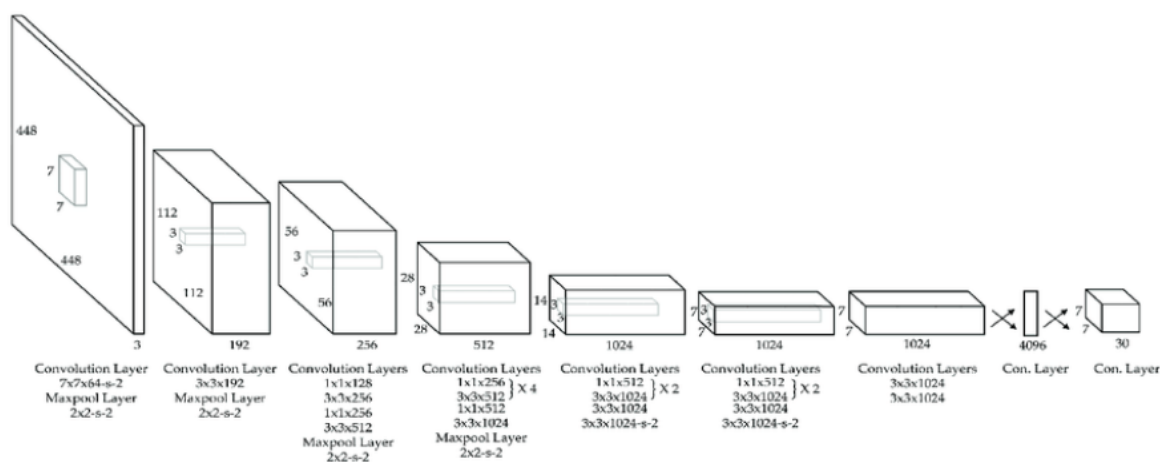


Obr. 2.9: SSD architektúra [15].

2.5.3 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 [16].

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



Obr. 2.10: YOLO architektúra [16].

2.6 Datasetsy

Na tréovanie neurónovej siete je dôležité mať k dispozícii dostatočné množstvo dát. 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.

Jedným z najpopulárnejších datasetov, ktorý slúži na rôzne úlohy počítačového videnia, je dataset COCO [17]. Ide o rozsiahly súbor údajov určených na detekciu, segmentáciu a popisovanie objektov. Aktuálne obsahuje viac ako 330 tisíc obrázkov, z ktorých je viac ako 200 tisíc anotovaných a obsahuje 80 tried objektov.

Ďalším veľkým a rovnako populárnym datasetom v oblasti počítačového videnia je ImageNet, ktorý je určený hlavne na úlohu rozpoznávania objektov. Tento dataset obsahuje viac ako 14 miliónov obrázkov, ktoré sú ručne anotované do viac ako 20 tisíc kategórii [18].

V oblasti dopravy existuje taktiež dostatočné množstvo dát. Medzi najpopulárnejšie datasety v tejto problematike patrí LISA dataset, ktorý sa zameriava na detekciu a rozpoznávanie dopravného značenia pri cestách. Obsahuje 6610 snímok z videí, na ktorých je anotovaných až 7855 objektov.



Obr. 2.11: Ukážka označených dopravných značiek z datasetu LISA [19].

2.7 Detekcia významných oblastí v obraze

Pri sledovaní okolitého sveta prichádza do ľudského oka množstvo vizuálnych informácií. Na spracovanie veľkého množstva údajov, ktoré do očí prichádzajú, je potrebné rozhodnúť, ktoré informácie sú najrelevantnejšie. Vďaka vizuálnej pozornosti dokáže ľudský mozog odfiltrovať nerelevantné informácie, potenciálne menej dôležité oblasti, a pri vizuálnom spracovaní používa údaje z významných oblastí scény. Pomocou vizuálnej pozornosti sú pohyby očí riadené tak, že pozornosť je priťahovaná k významným podnetom. Zaznamenávaním pohybu očí dokážeme pozorovať, čo priťahuje našu pozornosť.

V nedávnom období došlo k vývoju v modelovaní vizuálnej pozornosti. Nové prístupy a architektúry modelov sa snažia predpovedať, kam sa pozrieme alebo aké sú významné oblasti vizuálnej scény. Tieto modely sú hodnotené porovnaním so skutočným pohybom očí, ktorý je zaznamenaný.

Pojem významná oblasť vieme vysvetliť ako nejaké viditeľné lokálne zmeny v obraze. Pri významných oblastiach rozlišujeme dva typy významnosti. Pri nízko-úrovňovej významnosti je pozornosť podvedome priťahovaná k lokálnym zmenám v obraze. Pri vysoko-úrovňovej významnosti 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. Takýmto prípadom je napríklad červené jablko na zelenom strome alebo červené pery na ľudskej tvári. Procesy sú nezávislé na úlohe, ktorú pozorovateľ rieši a využívajú sa pri automatickej detekcii významných oblastí v scéne [20].
- 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 [20].

Sledovanie pohybu očí

Sledovanie pohybu očí je kľúčovou technológiou, ktorá napomáha pochopiť ľudské správanie a základné procesy myslenia. Aplikácie sledovania pohybu očí a pohľadu sú neobmedzené vo výskume i v komerčnom využití. Sledovanie pohybu očí umožňuje meranie pohybu očí, pozíciu očí a body pohľadu pomocou rôznych technologických procesov. Sledovanie pohybu očí identifikuje a monitoruje vizuálnu pozornosť sledovanej osoby z hľadiska polohy, objektov a trvania. Sledovanie pohybu očí môže byť využité napríklad na meranie frekvencie a pravidelnosti žmurkania, identifikáciu objektov, ktoré sledovaná osoba pozoruje/ignoruje, hodnotenie reakcie zreničky na rôzne podnety a vývoj interakcie človek - počítač a strojového učenia [21].

Oko vykonáva veľké množstvo pohybov, avšak z hľadiska sledovania pohybu očí sú najrelevantnejšie merané fixácie, sakády a sledovací pohyb. Fixácie sa objavujú, keď oko zastaví na zber vizuálnych údajov. Sakády predstavujú rýchle skoky, ktoré oko vykonáva medzi fixáciami v statickom prostredí. Sledovací pohyb oko vykonáva pri sledovaní pomaly sa pohybujúceho objektu [21].

Na detekciu a sledovanie očí sa používajú systémy sledovania očí. Existujú dva hlavné typy systémov sledovania pohybu očí:

- senzorové sledovanie očí,
- sledovanie očí založené na počítačovom videní.

Senzorové sledovanie očí

Niektoré systémy na sledovanie pohybu očí detegujú a analyzujú pohyby očí na základe elektrického potenciálu meraného elektródami, ktoré sú umiestnené v oblasti okolo očí. Elektródy pre každé oko merajú elektrické potenciálove pole pomocou elektrookulogramu. Pohyb oka potom možno sledovať analýzou zmien signálu elektrookulogramu [22].

Sledovanie očí založené na počítačovom videní

Techniky na detekciu očí a sledovanie pohľadu založené na počítačovom videní patria k najbežnejším prístupom v tejto oblasti výskumu. Na zaznamenávanie pohybu očí sa

používa kamera. Umiestnenie kamery na sledovanie očí závisí od cieľa, s ktorým sledujeme pohyb očí. Napríklad pri pozorovaní okolitého prostredia je vhodné, aby kamera bola umiestnená na hlave, v prípade sledovania obrázkov na monitore, je potrebné, aby kamera aj hlava boli v stabilnej polohe [23]. Na sledovanie polohy očí existuje mnoho rôznych techník. Medzi tieto techniky patria napríklad sledovanie očí pomocou rozpoznávania vzorov [24], sledovanie polohy očí založené na bodoch odrazu rohovky [25], sledovanie pohybu očí na základe tvaru [26], sledovanie polohy očí pomocou efektu tmavých a svetlých zreníc [27], pomocou modelu očí [22], prípadne pomocou hybridných techník sledovania polohy očí [28]. Techniky sledovania pohybu očí sa využívajú aj na detekciu významných oblastí.



Obr. 2.12: Model eyetrackera Tobii Glasses 3 [29].

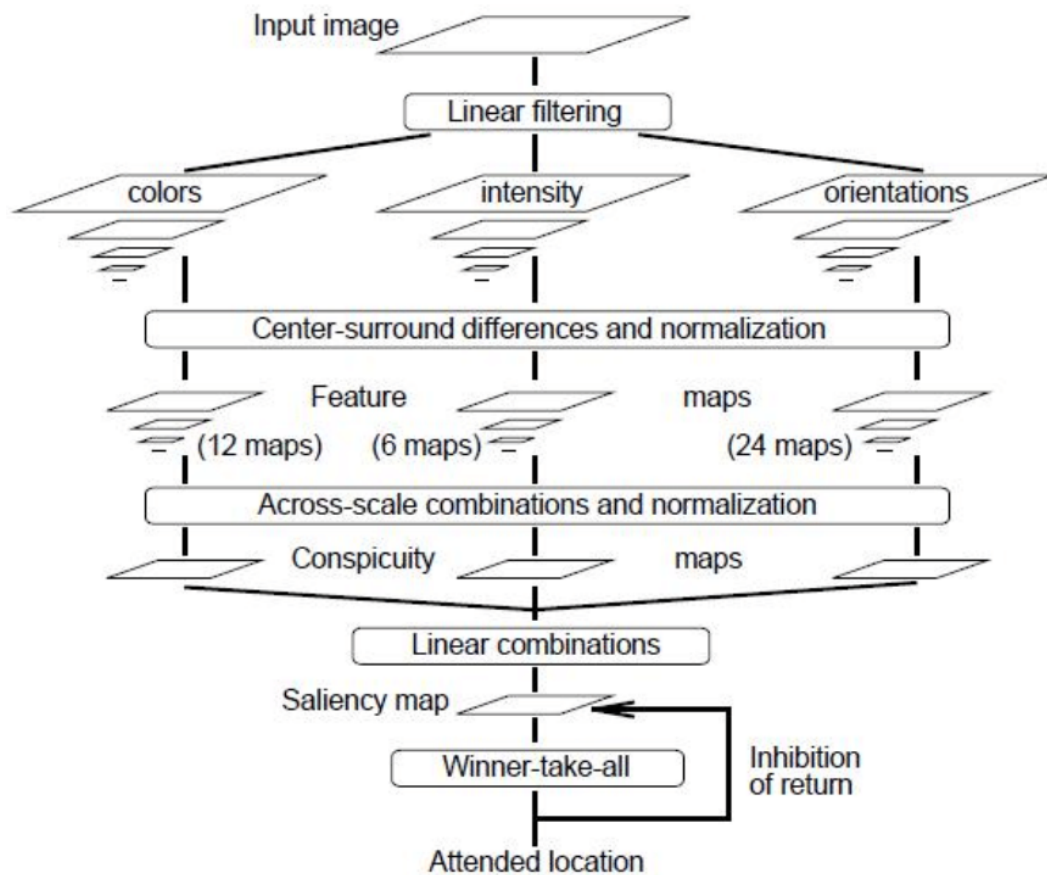
2.7.1 Metódy na detekciu významných oblastí

Medzi najstaršie oblasti výskumu tejto problematiky patria metódy pre statické obrazy, ktoré tvoria základ všetkých modelov na detekciu významností. V tejto časti si opíšeme najznámejší model pre výpočet významných oblastí - Ittiho model a metódu Spektrálne reziduá.

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 ob-

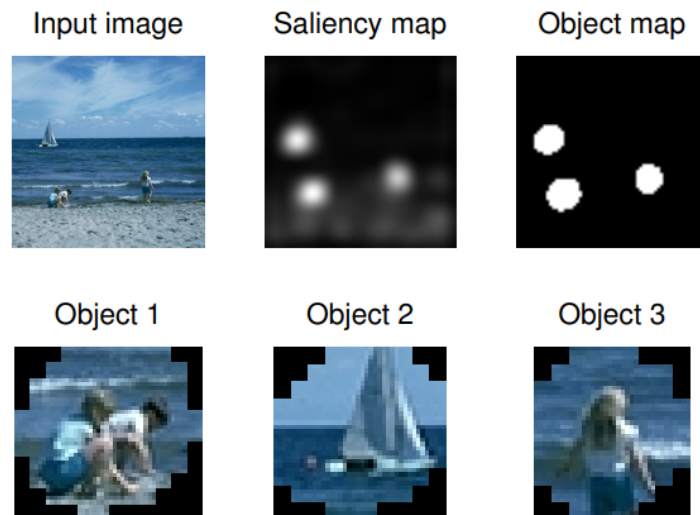
lastí (saliency map) a príznaky v obraze sa hľadajú v troch paralelných kanáloch pre farbu, intenzitu a orientáciu, kde sa hľadajú zmeny. Model celkovo vytvorí 42 máp, z ktorých 6 prislúcha intenzite, 12 farbe a 24 orientácii. Týchto 42 máp je následne lineárne skombinovaných do jednej mapy významnosti [30].



Obr. 2.13: Architektúra Ittiho modelu [6].

Spektrálne reziduá

Hlavnou myšlienkou tejto metódy je, že pomocou Fourierovej transformácie sa obrázok rozdelí na fázovú a amplitúdovú zložku. Vyhladením amplitúdovej zložky sa zvýraznia informácie obrázku, ktoré sú najviac vzdialené priemeru. Po odčítaní tejto vyhladenej amplitúdovej zložky od pôvodnej amplitúdovej zložky v obraze zostanú len významné oblasti, ktoré boli najviac vzdialené priemeru [31].



Obr. 2.14: Detekcia objektov na obrázku pomocou metódy spektrálnych reziduí [31].

2.8 Datasets

Veľmi populárnym datasetom, ktorý vznikol za účelom riešenia úloh detekcie významnosti, je dataset SALICON (The SALiency in CONtext). Pozostáva z tréningovej množiny (10 tisíc obrázkov), validačnej množiny (5 tisíc obrázkov) a testovacej množiny (5 tisíc obrázkov). Všetky obrázky pochádzajú z datasetu COCO a dataset obsahuje aj anotácie významných oblastí [32].

Pre účely detekcie významnosti vo videách bol vytvorený dataset DHF1K (Dynamic Human Fixation), ktorý pozostáva z tisíc vysoko kvalitných, vybraných videosekvencií, zahŕňajúcich širokú škálu rôznych scén, objektov, pohybov a zložitosti pozadia. Dataset zahŕňa binárne ground truth mapy fixácií očí [33].



Obr. 2.15: Obrázky z DHF1K datasetu. V prvom riadku sú pôvodné obrázky a v druhom riadku sú vyznačené oblasti fixácie očí [33].

Kapitola 3

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.

3.1 Detekcia reklamnej plochy

V tejto časti sa budeme zaoberať detekciou reklamnej plochy.

Dataset

Prvým krokom, ktorý budeme musieť riešiť, je výber vhodného datasetu. Keďže pri riešení budeme používať neurónové siete, na tréning bude potrebné veľké množstvo dát. V našom riešení budeme pracovať s dvoma datasetmi. Prvým je vlastný dataset, ktorý vytvoríme pomocou snímok získaných z palubnej dosky automobilu. Tvorba datasetu je veľmi náročný a zdĺhavý proces, pretože je potreba získať veľké množstvo snímok z ciest z mesta alebo mimo neho a následne tieto obrázky ručne anotovať. Ako druhý dataset použijeme už vytvorený voľne dostupný dataset. Pri výbere správneho datasetu musíme byť pozorný, aby obsahoval dostatočné množstvo snímok s objektami, ktoré budeme detegovať - teda reklamné plochy. Jedným z takýchto datasetov by mohol byť Mapillary Vistas Dataset [34]. Je to dataset vytvorený pre detekciu rôznych

objektov. Obsahuje snímky s vysokým rozlíšením, ku ktorým prislúcha priložený súbor s anotáciami objektov.

Model

Ďalším krokom bude výber modelu. Cieľom tohto kroku bude vybrať model, ktorý je dostatočne presný, ale aj rýchly. Pre riešenie tejto problematiky by mohol byť vhodný model YOLO, ktorý je momentálne veľmi populárny, pretože dosahuje veľmi dobré výsledky, čo sa týka rýchlosti za zachovania presnosti detekcie. Druhým modelom, ktorý sme si vybrali na otestovanie, je model Faster R-CNN. Pri riešení použijeme predtrénované modely, ktoré nám pomôžu s nastavením vstupných parametrov a hyperparametrov, čím výrazne skrátia čas tréningu. Modely následne vyhodnotíme a model s lepšími výsledkami použijeme pre získavanie oblastí reklamných plôch pre detekciu významnosti.

3.2 Detekcia významnosti reklamných plôch

V tejto časti budeme pracovať s oblasťami obrázku, ktoré sme získali pomocou modelov na detekciu objektov a ďalej budeme detegovať významnosť týchto získaných oblastí. Základným krokom pri detekcii významnosti bude vytvorenie mapy významností z pôvodného obrázku. Pre tento účel implementujeme a otestujeme dve metódy určené na detekciu významnosti. Prvá z týchto metód využíva spektrálne reziduá. Potom implementujeme a otestujeme model UniSal, ktorý využíva prístup hlbokého učenia [35]. UniSal patrí medzi state of the art algoritmy a dokáže pracovať so statickými obrázkami ale aj s videami. Z tohto dôvodu predpokladáme lepšie výsledky práve touto metódou.

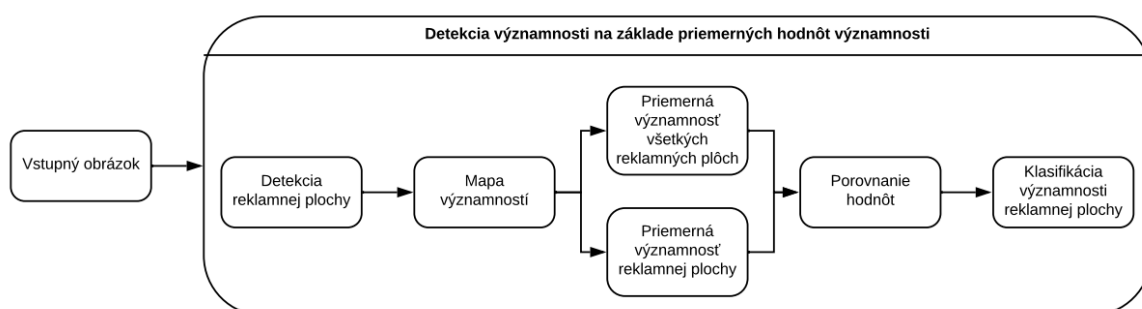
Po získaní máp významností implementujeme riešenie na detekciu významnosti reklamných plôch, ktoré je založené na základe priemernej hodnoty významnosti ohraničeného boxu reklamnej plochy.

Detekcia významnosti na základe hodnôt významnosti

Postup pri tomto riešení bude nasledovný:

1. detekcia reklamných plôch na vstupnom obrázku,

2. vytvorenie mapy významností pre vstupný obrázok,
3. získanie priemernej významnosti reklamných plôch,
4. kontrola, či je hodnota priemernej významnosti reklamnej plochy dostatočne veľká,
5. klasifikácia významnosti reklamných plôch.



Obr. 3.1: Návrh k postupu detekcie významnosti reklamnej plochy.

Na záver budeme významnosť reklamných plôch verifikovať na základe výstupu z eyetrackera. Ak bola reklamná plocha klasifikovaná ako významná, tak by sa na snímke z eyetrackera mala na reklamnej ploche nachádzať fixácia.

Kapitola 4

Riešenie problematiky a implementácia

Implementáciu aj testovanie sme vykonávali na počítači s procesorom AMD Ryzen 7 2700, ktorý pracuje na frekvenci 4,1 GHz s 32 GB RAM a s grafickou kartou GeForce RTX 2070. Implementáciu sme vykonávali v programovacom jazyku Python. Na verifikáciu bol použitý eyetracker Tobii Pro Glasses 3 a so získanými dátami sme pracovali v aplikácii Tobii Pro Lab.

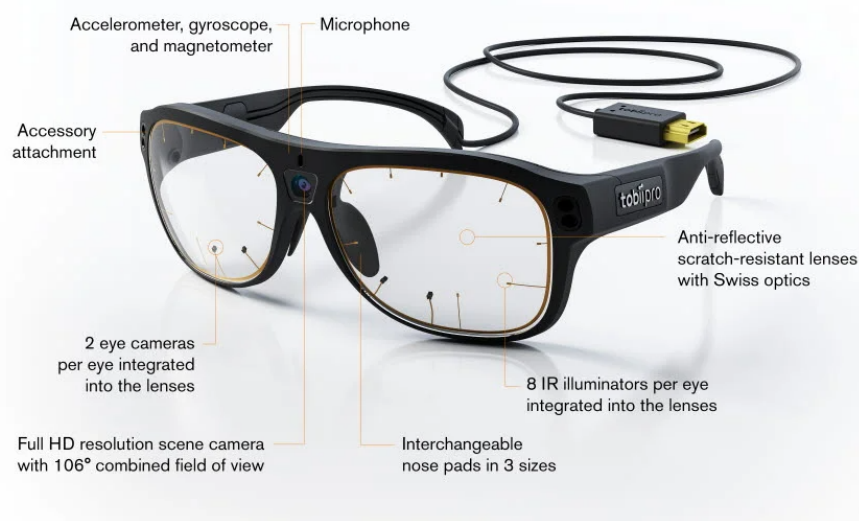
4.1 Pytorch

Pytorch je open-source knižnica, ktorá je založená na knižnici Torch a je určená primárne na účely strojového učenia (machine learning) a hĺbkového učenia (deep learning). Táto knižnica sa najčastejšie používa v kombinácii s jazykom Python. Okrem toho je možné ju použiť aj s jazykom C++. Je vyvíjaná hlavne výskumným tímom Facebooku pre umelú inteligenciu (Facebook's AI Research lab) [36].

Je jednoduchý na naučenie a používanie. Obsahuje obširnu dokumentáciu a rôzne tutoriály. Výhodou tejto knižnice je, že umožňuje rýchle počítanie tzv. tensorov (výstup je podobný matici) použitím GPU.

4.2 Eyetracker Tobii Pro Glasses 3

Okuliare Tobii Pro Glasses 3 sú navrhnuté tak, aby umožňovali jednoduchý, presný a efektívny zber údajov sledovania očí v širokej škále scenárov výskumu. Diskrétny, ultraľahký dizajn zaisťuje prirodzené správanie a validitu výskumu v kvalitatívnych aj kvantitatívnych situáciách. Na presné údaje a robustné funkcie sledovania očí sa možno spoľahnúť v nekontrolovaných situáciách a v reálnych prostrediach. Eyetracker obsahuje 16 iluminátorov a 4 očné kamery integrované do šošoviek odolných voči poškrabaniu, čo umožňuje optimálne umiestnenie a poskytuje užívateľovi ničím nerušený výhľad. Aplikácia pre eyetracker má podporu pre Windows, macOS aj Android. To znamená, že sa dá ovládať aj pomocou mobilného telefónu [29]. Okuliare Tobii Pro Glasses 3, ktoré boli použité pri riešení tejto práce boli zakúpené vďaka projektu Advancing University Capacity and Competence in Research, Development a Innovation (ACCORD).



Obr. 4.1: Eyetracker Tobii Glasses 3 [29].

4.3 Dataset

4.3.1 Mappilary Vistas Dataset

Tento 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. Testovacia množina neobsahuje anotácie objektov. Obrázky v datasete sú rozdelené do troch množín nasledovne:

- Trénovacia množina - 18 tisíc obrázkov,
- Validačná množina - 2 tisíc obrázkov,
- Testovacia množina - 5 tisíc obrázkov.



Obr. 4.2: Ukážky ohraničujúcich boxov reklamných plôch v Mapillary Vistas Datasete.

4.3.2 Vlastný dataset

Tento dataset obsahuje 3000 obrázkov, ktoré boli získané pomocou mobilného telefónu na palubnej doske automobilu a eyetrackera na hlave vodiča automobilu. Pre všetky obrázky datasetu sme zvolili rovnaké rozlíšenie 1920×1080 pixelov. Všetky obrázky boli ručne anotované tak, že každému obrázku prislúcha jeden súbor. Obrázky v datasete sme rozdelili na trénovaciu a validačnú množinu nasledovne:

- Trénovacia množina - 1350 obrázkov,
- Validačná množina - 150 obrázkov.



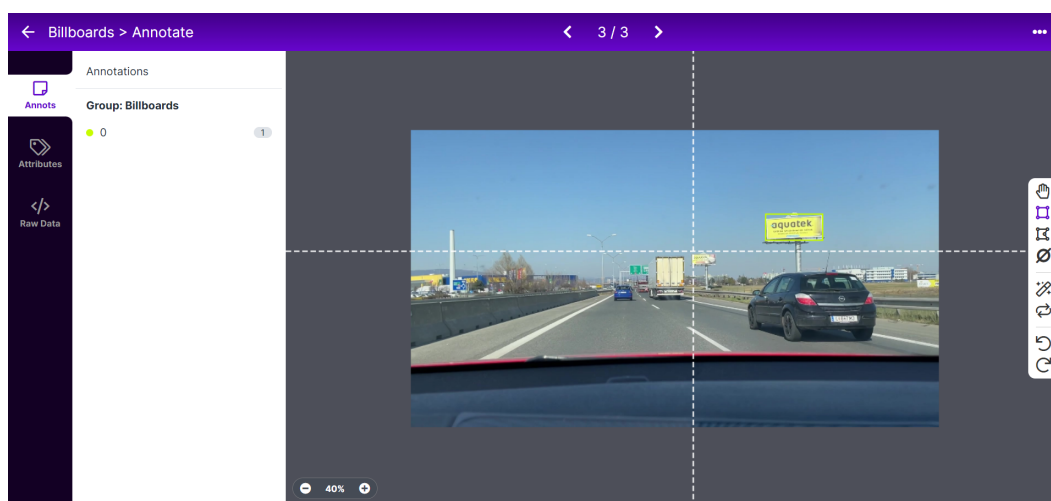
Obr. 4.3: Ukážky ohraničujúcich boxov reklamných plôch vo vlastnom datasete.

4.3.3 Príprava datasetov a súborov

V tejto sekcii sa venujeme tvorbe datasetu a príprave súborov.

Anotovanie reklamných plôch

Pri tvorbe vlastného datasetu bolo potrebné na jednotlivých obrázkoch označiť reklamné plochy, ktoré budeme detegovať. Na riešenie tohto problému bol použitý Roboflow. Ide o platformu počítačového videnia, ktorá používateľom umožňuje rýchlejšie a presnejšie vytvárať modely počítačového videnia prostredníctvom poskytovania lepšieho zberu údajov, predspracovania a techník tréningu modelov. Okrem toho používateľom umožňuje nahrávať vlastné súbory údajov, kresliť anotácie, upravovať orientáciu obrázkov, meniť veľkosť obrázkov, upravovať kontrast obrázkov a vykonávať rozširovanie údajov [37].



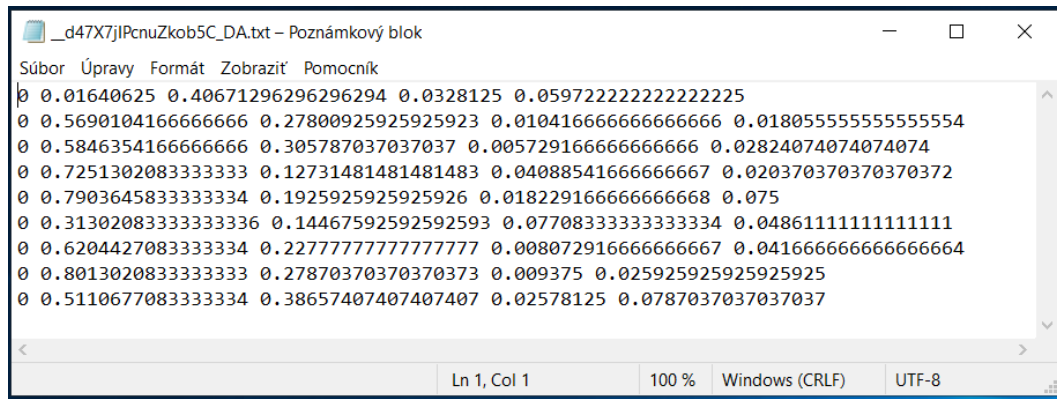
Obr. 4.4: Anotovanie reklamných plôch v prostredí Roboflow.

Pri anotácii reklamných plôch bolo potrebné zvoliť spôsob, akým budeme reklamné plochy anotovať, nakoľko v datasete sa nachádzajú rôzne situácie, kde je reklamná plocha čiastočne alebo vo veľkej miere prekrytá iným objektom. Takéto prípady sme riešili čo najpresnejším označením reklamnej plochy aj na miestach, kde bola prekrytá. Ďalším problémom bol prípad, ak je reklamná plocha vo veľmi veľkej vzdialenosti a nedokážeme určiť ľudským okom, či ide v skutočnosti o reklamnú plochu alebo dopravné značenie. Preto sme takéto prípady nezahrnuli do nášho datasetu. Pretože naša detekcia sa zameriavala primárne na veľké reklamné plochy - billboardy, tak sme z datasetu vylúčili menšie reklamné plochy, ako napríklad názvy obchodov, logá firiem, atď.



Obr. 4.5: Ukážka anotovaných prekrytých reklamných plôch.

Pred začiatkom trénovania sme museli pripraviť potrebné súbory datasetov. Model YOLO vyžaduje textový súbor s anotáciami objektov s rovnakým názvom ako príslušný obrázok. V jednom riadku tohto textového súboru sa nachádzajú informácie o jednom 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. Pre model Faster R-CNN sme pripravili dáta v .json tak, že každému obrázku prislúchal jeden .json súbor.



Obr. 4.6: Príklad textového súboru s anotáciami objektov.

4.4 Transfer learning

Pre zjednodušenie a skrátenie času trénovania modelov sme si zvolili a použili už predtrénované modely. Veľkou výhodou takýchto modelov je, že sú predtrénované na veľkých datasetoch, a preto na dotrénovanie modelu, pri ktorom sa doučí nové informácie z nových dát, už nie je potreba veľkého množstva dát. Takýto prístup je v dnešnej dobe veľmi populárny a v danej problematike sa nazýva transfer learning alebo prenášané učenie. Transfer learning je proces, pri ktorom sa model, ktorý bol natrénovaný na riešenie problému jednej úlohy, použije na riešenie problému inej úlohy. V súčasnosti máme k dispozícii veľa predtrénovaných modelov, ktoré sú natrénované na rôznych veľkých datasetoch, ako napríklad COCO, ImageNet, Kitti a ďalšie.

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 verziu YOLOv5, ktorá je voľne dostupná na GitHub [38], kde máme k dispozícii na stiahnutie aj predtrénované checkpointy na datasete COCO. K dispozícii je tam 5 rôznych verzií checkpointov - YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x. Na GitHub sa k týmto checkpointom nachádza tabuľka, v ktorej sú uvedené rôzne metriky (napr. presnosť, mAP, rýchlosť) a užívatelia si tak môžu vybrať najvhodnejší checkpoint podľa potreby.

Model	size (pixels)	mAP ^{val} 0.5:0.95	mAP ^{val} 0.5	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.0	45.7	45	6.3	0.6	1.9	4.5
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	1.7	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7
YOLOv5n6	1280	36.0	54.4	153	8.1	2.1	3.2	4.6
YOLOv5s6	1280	44.8	63.7	385	8.2	3.6	12.6	16.8
YOLOv5m6	1280	51.3	69.3	887	11.1	6.8	35.7	50.0
YOLOv5l6	1280	53.7	71.3	1784	15.8	10.5	76.8	111.4
YOLOv5x6	1280	55.0	72.7	3136	26.2	19.4	140.7	209.8
+ TTA	1536	55.8	72.7	-	-	-	-	-

Obr. 4.7: Zoznam predtrénovaných checkpointov modelu YOLOv5 s vyhodnotením, ktoré sú dostupné na GitHub.

Faster R-CNN

Ako druhý model sme si zvolili model Faster R-CNN, ktorý sme implementovali pomocou knižnice Detectron2, čo je knižnica od Facebook AI Research, ktorá poskytuje najmodernejšie algoritmy detekcie a segmentácie objektov [39].

Name	lr sched	train time (s/iter)	inference time (s/im)	train mem (GB)	box AP	model id	download
R50-C4	1x	0.551	0.102	4.8	35.7	137257644	model metrics
R50-DC5	1x	0.380	0.068	5.0	37.3	137847829	model metrics
R50-FPN	1x	0.210	0.038	3.0	37.9	137257794	model metrics
R50-C4	3x	0.543	0.104	4.8	38.4	137849393	model metrics
R50-DC5	3x	0.378	0.070	5.0	39.0	137849425	model metrics
R50-FPN	3x	0.209	0.038	3.0	40.2	137849458	model metrics
R101-C4	3x	0.619	0.139	5.9	41.1	138204752	model metrics
R101-DC5	3x	0.452	0.086	6.1	40.6	138204841	model metrics
R101-FPN	3x	0.286	0.051	4.1	42.0	137851257	model metrics
X101-FPN	3x	0.638	0.098	6.7	43.0	139173657	model metrics

Obr. 4.8: Zoznam predtrénovaných checkpointov modelu Faster R-CNN s vyhodnotením, ktoré sú dostupné na GitHub.

4.4.1 Trénovanie modelov

Počas trénovania modelu sa používajú dve množiny - trénovacia a validačná, na ktorých sa náš model trénuje a zároveň testuje. Pre každý tréning sme vopred určili vstupné parametre a hyperparametre. Pri trénovaní sme sa snažili hyperparametre nastaviť čo najoptimálnejšie, aby náš model dosiahol čo najlepšie výsledky.

Počas trénovania bolo dôležité sledovať, aby chyba trénovania klesala a v opačnom prípade je potrebné tréning včas ukončiť. Ak by sme pokračovali v trénovaní aj v prípade, že chyba už neklesá, tak by mohlo nastať pretrénovanie modelu (overfitting).

Po ukončení každej epochy sa nám aktualizované váhy uložili do súboru, takže ich vieme použiť ako checkpoint, z ktorého môžeme ďalej pokračovať v trénovaní.

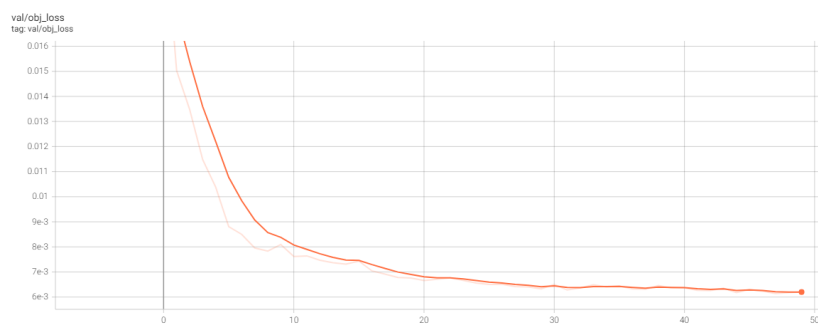
Z dostupných checkpointov pre model YOLOv5 sme použili a otestovali checkpoint s označením "YOLOv5m6", ktorý môžeme vidieť v tabuľke na obrázku 4.7. Pre otestovanie modelu Faster R-CNN sme vybrali z modelovej zoo na obrázku 4.8 model s označením "R101-FPN". Tento model používa okrem siete Faster R-CNN aj siete ResNet (Residual Neural Network) a FPN (Feature Pyramid Network), ktoré sa používajú na extrakciu príznakov. Oba modely sú predtrénované na datasete COCO, ktorý v anotáciách nezahŕňa triedu reklamné plochy, a preto sme modely dotrénovali na vlastnom, respektíve Mapillary Vistas datase.

Pri hľadaní optimálneho nastavenia pre získanie najlepších hodnôt tréningu sme experimentovali s týmito vstupnými parametrami a hyperparametrami: veľkosť vstupných obrázkov, krok učenia (learning rate), počet epôch/iterácií, veľkosť dávky (batch size), augmentácia dát. Pri voľbe hodnôt pre parametre, ako veľkosť vstupných obrázkov a veľkosť dávky, sme museli brať ohľad na dostupný hardvér a volili sme najväčšie možné hodnoty.

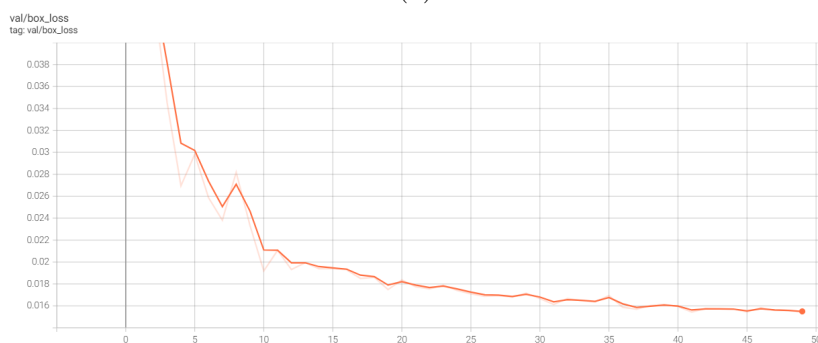
Sledovanie strát počas trénovania

Počas trénovania sme mali k dispozícii kontrolný výpis, v ktorom sa nachádzajú rôzne informácie, ako napríklad hodnoty AP, celkovú stratu, presnosť, citlivosť a ďalšie. Okrem tohto výpisu sme mali k dispozícii nástroj Tensorboard, v ktorom sme mohli sledovať grafy týchto metrík. Tieto grafy sa priebehom času aktualizujú. Najdôležitejším grafom, ktorý sme sledovali, je graf celkovej straty. Zo začiatku trénovania celková strata dosahovala vysoké hodnoty, ktoré by mali pri dobrom trénovaní postupne kle-

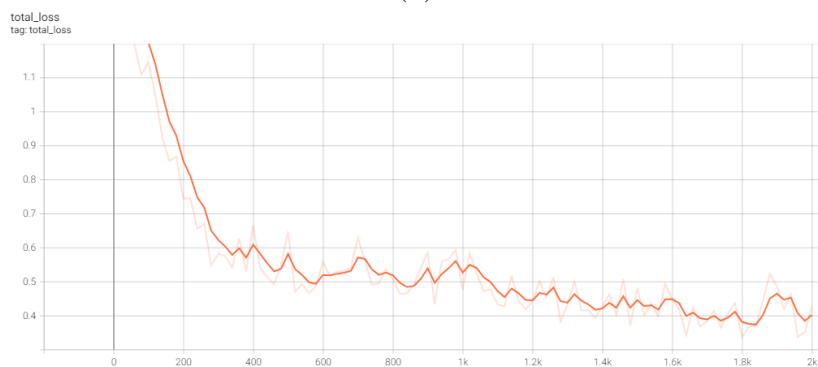
sať. Čím menšiu hodnotu straty model pri tréňovaní dosiahne, tým je model lepší. V prípade, ak by strata prestala klesať, bolo by potrebné tréňovanie ukončiť, aby nedošlo k pretréňovaniu.



(a)



(b)



(c)

Obr. 4.9: Ukážka grafov chybových funkcií.

4.5 Významnosť reklamnej plochy

V tejto časti sme sa venovali implementácii a testovaniu metód na detekciu významnosti reklamných plôch. Základným krokom je tvorba mapy významností (saliency map).

4.5.1 OpenCV Saliency API

Ako prvú sme implementovali metódu *spectralresidual*. Táto metóda je k dispozícii v knižnici OpenCV v časti Saliency API. Metóda *spectralresidual* využíva spektrálne reziduá a princíp fungovania tohto algoritmu je popísaný v časti 2.7.1. Výsledkom tejto metódy je mapa významnosti - šedotónový obrázok, v ktorom sa do popredia dostanú významné oblasti obrázku.

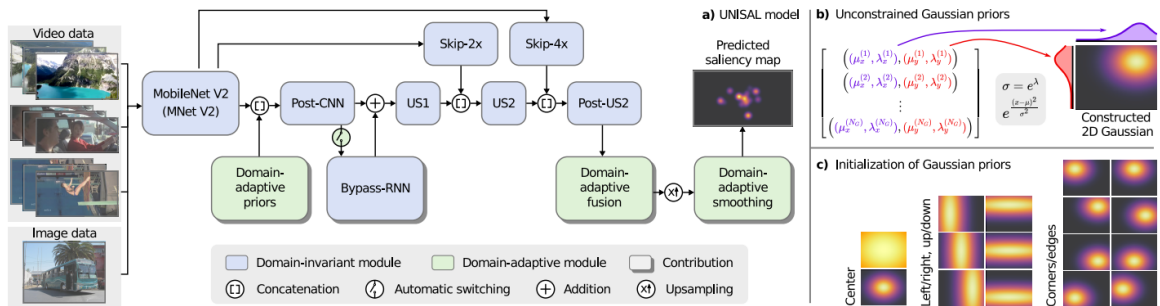
4.5.2 UniSal

Pri implementácii sme použili model predtrénovaný na DHF1K Datasets, ktorý je voľne dostupný na GitHub [40].

Tabuľka 4.1: Vyhodnotenie modelu UniSal na datasete DHF1K.

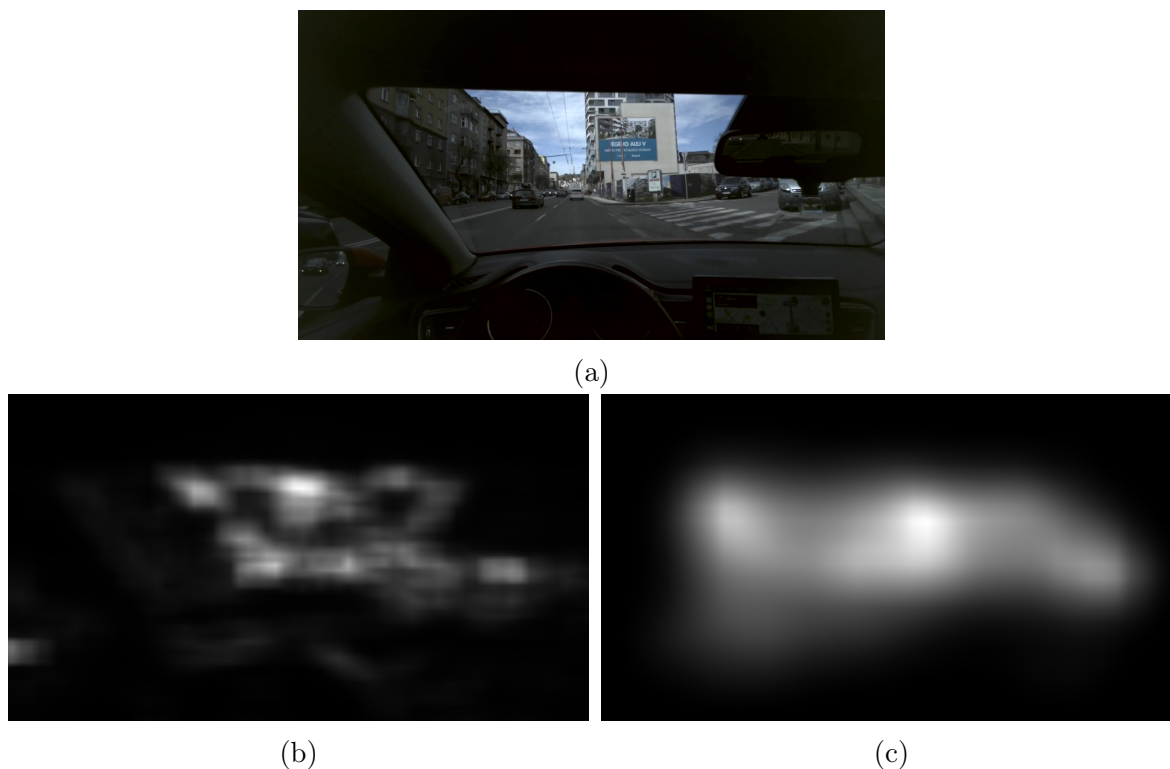
Dataset	AUC-J	SIM	s-AUC	CC	NSS	Čas (s)
DHF1K	0,901	0,390	0,691	0,490	2,776	0,009
Hollywood-2	0,934	0,542	0,795	0,673	3,901	-
UCF Sports	0,918	0,523	0,755	0,644	3,381	-

V tabuľke 4.1 môžeme vidieť vyhodnotenie modelu UniSal na datasetoch pre detekciu významnosti vo videách pomocou rôznych metrík.



Obr. 4.10: Architektúra UniSal.

Architektúra modelu UniSal je jednoduchá a navrhnutá tak, aby súčasne modelovala významnosť na obrázku aj videu. Architektúra využíva dizajn encoder - RNN - decoder, ktorý je prispôsobený pre modelovanie významnosti. Ako Encoder sieť sa používa MobileNet-V2 (MNet V2). Ako Bypass RNN sa, kvôli jednoduchosti, používa konvolučná GRU (Gated Recurrent Unit) RNN (Recurrent Neural Network).



Obr. 4.11: Mapy významností vytvorených metódami *spectralresidual* a *UniSal*.

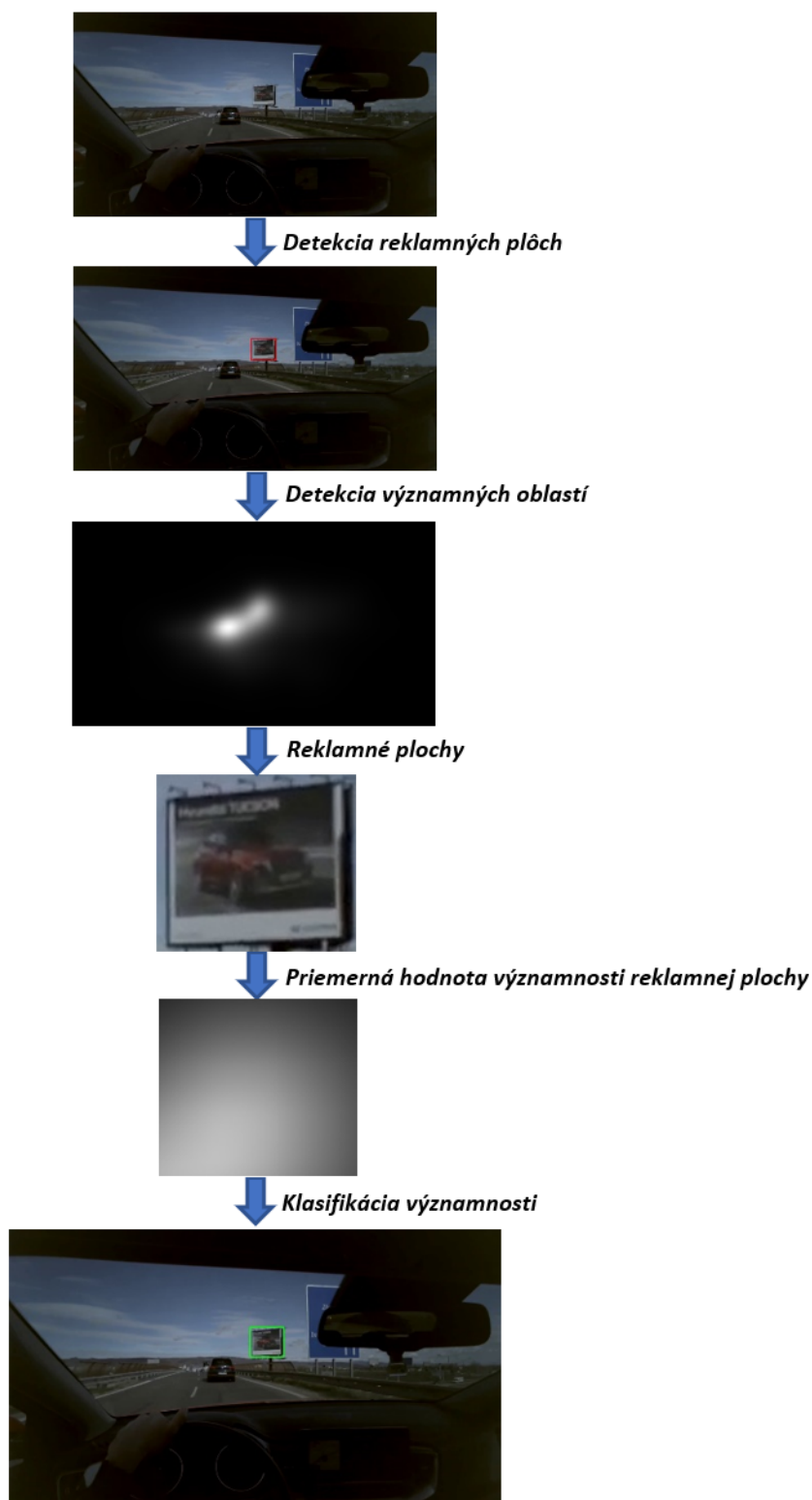
4.5.3 Detekcia významnosti reklamnej plochy

Pre určenie významnosti reklamnej plochy sme implementovali riešenie, ktoré je opísané v návrhu riešenia v kapitole 3.2 a je založené na základe priemernej hodnoty významnosti ohraničeného boxu reklamnej plochy.

Detekcia významnosti na základe hodnôt významnosti

Pri implementácii tohto riešenia sme postupovali podľa navrhnutého postupu v kapitole 3.2. Najskôr sme pomocou objektového detektora získali oblasti reklamných plôch. Potom sme metódami *spectralresidual* a *UniSal* vytvorili mapu významností, na ktorej sme pracovali už len s oblasťami reklamných plôch. V týchto oblastiach sme hodnoty významností všetkých pixelov normalizovali na interval od 0 do 1 a tieto hodnoty sme sčítali a predelili počtom pixelov v oblasti, čím sme získali priemernú hodnotu významností v danej oblasti. Ešte predtým sme rovnakým spôsobom získali priemernú hodnotu významností všetkých reklamných plôch. Pri klasifikovaní významnosti reklamnej plochy sme porovnali priemernú hodnotu významností reklamnej plochy a priemernú hodnotu významností všetkých reklamných plôch a ak bola väčšia, tak reklamná plo-

cha bola klasifikovaná ako významná. Ak sa na obrázku nachádzalo viacero reklamných plôch klasifikovaných ako významné, tak zostala plocha s najväčšou hodnotou priemernej významnosti a ostatné reklamné plochy boli klasifikované ako nevýznamné.



Obr. 4.12: Postup detekcie významnosti reklamných plôch zobrazený na obrázkoch.

Kapitola 5

Vyhodnotenie a výsledky

Vyhodnotenie prebiehalo v dvoch fázach. V prvej sme sa zamerali na vyhodnotenie a porovnanie modelov YOLOv5 a Faster R-CNN, ktorých úloha je detekcia reklamných plôch. V tej druhej sme vyhodnotili metódy na detekciu významnosti v obraze a metódu na detekciu významnosti reklamných plôch.

Pre účel testovania sme vytvorili testovací dataset, ktorý pozostáva zo 80 statických obrázkov, ktoré celkovo obsahujú 210 reklamných plôch. Dataset obsahuje obrázky s veľkosťou 1920×1080 s reklamnými plochami z diaľnice aj miest v rôznych svetelných podmienkach, v rôznom počasí, s rôznym počtom reklamných plôch, ktoré môžu byť aj prekryté inými objektami.

5.1 Detekcia reklamnej plochy

5.1.1 Použité metriky

Pri klasifikácii objektov môžu nastať tieto prípady:

- True positive (TP) - reklamná plocha bola správne označená tam, kde sa v skutočnosti nachádza,
- True negative (TN) - reklamná plocha správne nebola označená,
- False positive (FP) - reklamná plocha bola označená tam, kde sa v skutočnosti nenachádza,

- False negative (FN) - reklamná plocha nebola označená tam, kde sa v skutočnosti nachádza.

Precision (Presnosť)

Presnosť sa určuje ako pomer medzi počtom pravdivo pozitívnych prípadov a počtom všetkých pozitívnych prípadov. Táto hodnota určuje, do akej miery bola predikcia daného detektora správna.

$$Precision = \frac{TP}{TP + FP}. \quad (5.1)$$

Presnosť dosahuje hodnoty na intervale 0 až 1, pričom čím je táto hodnota vyššia, tým je väčšia pravdepodobnosť, že všetky predikované pozitívne detekcie boli naozaj správne.

Recall (Citlivosť)

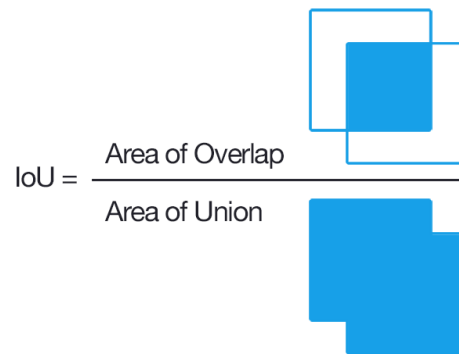
Citlivosť sa určuje ako pomer medzi počtom pravdivo pozitívnych a počtom všetkých skutočných vzorových objektov. Táto hodnota určuje, ako je daný detektor schopný detegovať všetky objekty na obrázku.

$$Recall = \frac{TP}{TP + FN}. \quad (5.2)$$

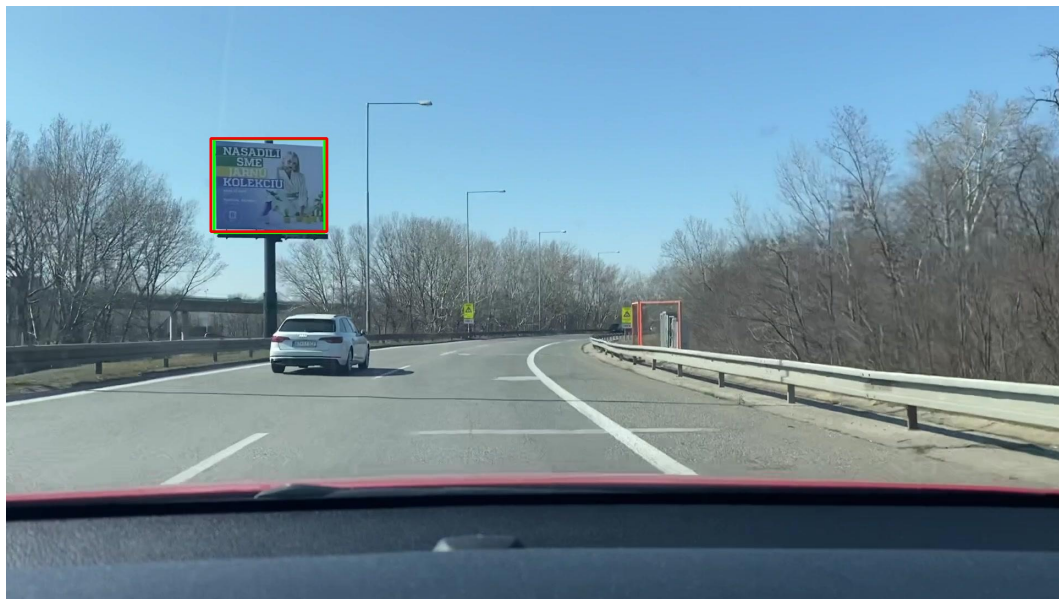
Citlivosť dosahuje hodnoty na intervale 0 až 1, pričom čím je táto hodnota vyššia, tým je väčšia pravdepodobnosť, že boli zdetekované všetky objekty na obrázku.

Intersection over Union - IoU

IoU je jedna z najdôležitejších metrík, ktorá sa používa na určenie presnosti detektora. Vyjadruje, ako veľmi sa prekrýva ohraničujúci box predikovaného objektu so skutočným ohraničujúcim boxom objektu, čo je ohraničujúci box, kde sa objekt reálne nachádza. Táto metrika dosahuje hodnoty na intervale 0 až 1, pričom čím je táto hodnota vyššia, tým je predikovaný ohraničujúci box presnejší.



Obr. 5.1: Metrika Intersection over Union

Obr. 5.2: Ukážka detekcie s hodnotou $\text{IoU} = 94\%$.

Average Precision - AP (Priemerná presnosť)

Jednou z najpoužívanějších a najpresnejších metrík na vyhodnotenie modelov určených na detekciu objektov je metrika označovaná ako AP. Používa sa aj pri súťažiach COCO a ide o metriku, ktorá vyjadruje priemer cez prahové hodnoty IoU od hodnoty 0,5 po 0,95 s krokom 0,05. Ďalšou používanou metrikou, je metrika Pascal VOC. Pri vyhodnocovaní touto metrikou sa predikovaný ohraničujúci box pokladá za skutočne pozitívny iba ak $\text{IoU} \geq 0,5$ v porovnaní so skutočným ohraničujúcim boxom objektu. Average precision je metrika, ktorá slúži na celkové vyhodnotenie detektora. Táto metrika vo výsledku zahŕňa presnosť aj citlivosť.

5.1.2 Experimenty

V tejto časti sú uvedené vykonané experimenty s najlepšimi dosiahnutými výsledkami. Model YOLOv5 dosiahol predikčný čas 0.058 sekundy na jeden obrázok veľkosti 1920×1080 , zatiaľčo model Faster R-CNN bol pomalší a jeho predikčný čas bol 0.513 sekundy na jeden obrázok.

Dataset Mapillary Vistas

Tabuľka 5.1: Vyhodnotenie modelov pri detekcii reklamných plôch po dotrénovaní na Mapillary Vistas datasete.

Model	AP@0.5	AP@0.5:0.95
<i>YOLOv5</i>	63,2%	48,1%
<i>Faster R-CNN</i>	55,8%	40,2%

V tabuľke 5.1 môžeme vidieť výsledky modelov po dotrénovaní na Mapillary Vistas datasete. Model Faster R-CNN bol dotrénovaný na 2000 iteráciách s veľkosťou dávky 4 obrázky, čo bolo maximum, ktoré nám náš hardvér dovolil. Model YOLO bol dotrénovaný na 50 iteračných epochách s veľkosťou dávky tiež 4 obrázky.

Tabuľka 5.2: Vyhodnotenie modelov pri detekcii reklamných plôch po dotrénovaní na Mapillary Vistas datasete a po pridaní augmentácie dát.

Model	AP@0.5	AP@0.5:0.95
<i>YOLOv5</i>	58,7%	47,0%
<i>Faster R-CNN</i>	54,1%	42,3%

Druhým experimentom, ktorý sme robili s týmto datasetom, bolo pridanie augmentácie dát. Postupne sme testovali rôzne kombinácie techník augmentácie dát a najlepšie výsledky sme získali pridaním nasledovných: náhodné otočenie (od -10° po 10°), náhodný jas (od -10% po 10%), náhodný šum a náhodné orezanie. V tabuľke 5.2 môžeme vidieť, že k výraznejšiemu zlepšeniu výsledkov nedošlo.

Vlastný dataset

Ďalej bolo vykonané testovanie modelov dotrénovaných na našom vlastnom datasete.

Tabuľka 5.3: Vyhodnotenie modelov pri detekcii reklamných plôch po dotrénovaní na vlastnom datasete.

Model	AP@0.5	AP@0.5:0.95
<i>YOLOv5</i>	96,6%	69,5%
<i>Faster R-CNN</i>	89,6%	58,4%

V tabuľke 5.3 môžeme vidieť výsledky modelov po dotrénovaní na vlastnom datasete. Model Faster R-CNN bol dotrénovaný na 2000 iteráciách s veľkosťou dávky 4 obrázky, rovnako ako pri prvom datasete. Model YOLO bol dotrénovaný na 40 iteračných epochách s veľkosťou dávky 4 obrázky.

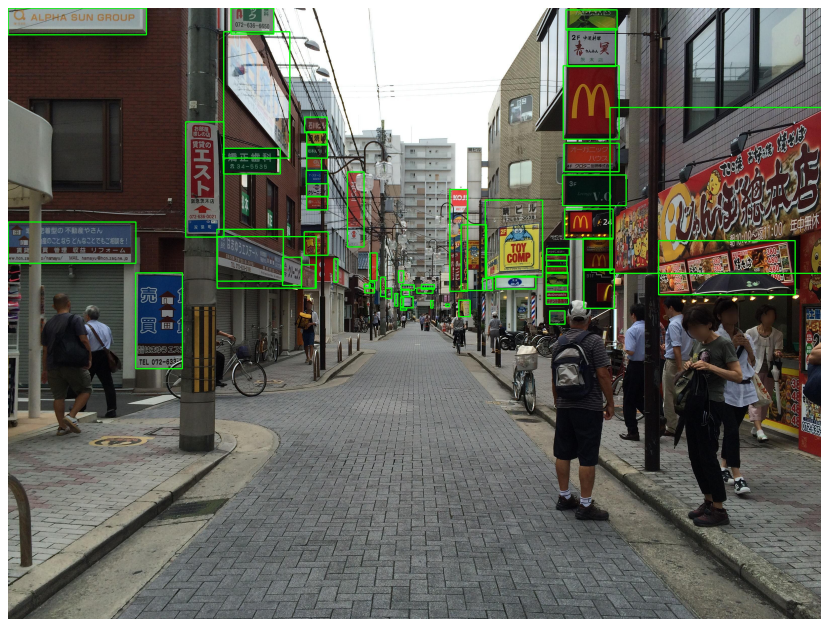
Tabuľka 5.4: Vyhodnotenie modelov pri detekcii reklamných plôch po dotrénovaní na vlastnom datasete a po pridaní augmentácie dát.

Model	AP@0.5	AP@0.5:0.95
<i>YOLOv5</i>	93,2%	65,4%
<i>Faster R-CNN</i>	85,3%	58,7%

Podobne ako pri experimentovaní s Mapillary Vistas datasetom sme otestovali vplyv pridania techník augmentácie dát aj na náš dataset. Pri testovaní sme použili totožné techniky augmentácie dát. V tabuľke 5.4 môžeme vidieť, že k výraznejšiemu zlepšeniu výsledkov taktiež nedošlo.

5.1.3 Porovnanie výsledkov

V tabuľkách vyhodnotenia si môžeme všimnúť, že na vlastnom datasete boli dosiahnuté výrazne lepšie výsledky ako na Mapillary Vistas datasete. Dôvodom týchto rozdielov je, že v Mapillary Vistas datasete sa nachádzajú, v porovnaní s našim datasetom, obrázky s oveľa vyšším rozlíšením (najväčšie až 5248×3936 pixelov), na ktorých sú anotované aj veľmi malé a vzdialené reklamné plochy. Pri tréňovaní modelov na detekciu objektov sa pri vstupe veľkosť obrázkov redukuje na menšiu (v našom prípade 1920×1080 pixelov), a tým dochádza k "zaniknutiu" veľmi malých reklamných plôch anotovaných na obrázkoch s vysokým rozlíšením.



Obr. 5.3: Anotované malé objekty v Mapillary Vistas datasete.



Obr. 5.4: Anotované malé objekty v Mapillary Vistas datasete.

Na obrázkoch 5.3 a 5.4 môžeme vidieť príklad veľmi malých anotovaných reklamných plôch popri ceste. Ďalším dôvodom horších výsledkov je, že vlastný dataset obsahuje aj obrázky snímané eyetrackerom, zatiaľčo Mapillary Vistas dataset obsahuje snímky zosnímané kamerou umiestnenou na streche auta. Problém pri detekcii mu mohol spôsobiť interiér auta, ako aj čelné sklo automobilu.

5.2 Detekcia významnosti

5.2.1 Použité metriky

Metriky určené na vyhodnocovanie presnosti modelov pre detekciu významnosti možno rozdeliť do troch skupín:

- založené na hodnotách - NSS, Percentile,
- založené na polohe - AUC-Judd, AUC-Zhao, AUC Borji, AUC-Li,
- založené na distribúcii.

V našom riešení použijeme na vyhodnotenie našich metód, metriky AUROC ((Area under the Receiver Operating Characteristic curve), konkrétnejšie AUC-Judd a NSS (Normalized Scanpath Saliency).

AUC-Judd

AUC-Judd metrika založená na polohe vychádzajúca z teórie detekcie signálu. Krivka ROC (Receiver Operating Characteristic curve) znázorňuje podiel falošne pozitívnych voči skutočne pozitívnym pre všetky možné prahové hodnoty. Plocha (oblasť) pod krivkou ROC sa vo výskume významnosti bežne používa a v priebehu rokov bolo vyvinutých veľa rôznych implementácií klasického prístupu. V tejto práci je použitá implementácia Judd [41]. Mapa významnosti je normalizovaná pomocou funkcie škálovania Min-Max

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}, \quad (5.3)$$

kde x' predstavuje normalizovanú hodnotu a x pôvodnú hodnotu. Pre každý prah (threshold) sa vypočíta miera skutočnej pozitivity (TP) a miera falošnej pozitivity (FP). Skutočne pozitívne sú významné mapové hodnoty nad prahom v miestach fixácii a miera TP predstavuje pomer skutočnej pozitivity k počtu všetkých fixácií

$$TP = \frac{TP}{\text{všetky fixácie}}. \quad (5.4)$$

Falošne pozitívne sú všetky ostatné významné mapové hodnoty nad prahom nachádzajúce sa mimo miest fixácií. Pomer FP je pomer počtu falošne pozitívnych k počtu

všetkých pixelov mapy významnosti mínus všetky fixácie

$$FP = \frac{FP}{\text{všetky pixely mapy významnosti} - \text{všetky fixácie}}. \quad (5.5)$$

Následne je možné ROC krivku vykresliť a konečné skóre predstavuje plocha pod ňou, ktorá je vypočítaná pomocou lichobežníkového pravidla (technika na aproximáciu oblasti pod grafom funkcie). V dokonalom prípade, kedy by mapa významnosti správne predpovedala všetky a iba reálne fixácie, skóre plochy pod krivkou by bolo 1. Náhodná klasifikácia by mala skóre 0,5 ($TP = FP$), preto sa skóre modelu bude pohybovať medzi 0,5 a 1.

NSS

NSS metrika je ďalšou metrikou založenou na polohe, ktorá je použitá v tejto práci. Táto metrika meria zhodu medzi mapou významnosti a mapou fixácie ako priemernú významnosť, vypočítanú zo štandardizovaných hodnôt mapy významnosti na miestach fixácie pozdĺž skenovacej trasy subjektov. Spomedzi množstva verzií NSS, ktoré sú dostupné, bola pre túto prácu vybratá implementácia Bylinskii [41]. V prvom kroku sú hodnoty významnosti štandardizované tak, aby mali nulovú strednú hodnotu a jednotkovú štandardnú odchýlku

$$Y_{SM}(x) = \frac{SM(x) - \mu}{\sigma}, \quad (5.6)$$

kde SM je mapa významnosti (Saliency Map), Y_{SM} je štandardizovaná mapa významnosti, μ je stredná hodnota vyjadrená ako

$$\mu = \frac{1}{|I|} \sum_{t \in I} SM(x_t) \quad (5.7)$$

a σ je štandardná odchýlka

$$\sigma = \sqrt{\frac{1}{|I|} \sum_{t \in I} (SM(x_t) - \mu)^2}, \quad (5.8)$$

kde $|I|$ je počet pixelov v obrázku. Hodnota $Y_{SM}(x_i)$ reprezentuje rozdiel hodnoty mapy významnosti x_i od priemernej hodnoty v jednotkách štandardnej odchýlky. Je pozi-

tívna, keď hodnota významnosti v mieste fixácie je nad strednou hodnotou a negatívna, keď je nižšia ako stredná hodnota. Celkové NSS skóre je potom dané vzťahom

$$NSS = \frac{1}{N} \sum_i Y_{SM}(x_i) \times Q_i, \quad (5.9)$$

kde N je počet fixácií a Q je binárna mapa fixácií. Je to stredná hodnota všetkých štandardizovaných hodnôt významnosti v miestach fixácií. Napríklad skóre 1 znamená, že hodnoty významnosti v mieste fixácie boli 1 jednotku štandardnej odchýlky nad priemernou hodnotou významnosti. Narozdiel od AUC, NSS pracuje so skutočnými hodnotami významnosti a je citlivejšia na falošnú pozitivitu.

5.2.2 Porovnanie metód na detekciu významnosti

Na porovnanie metód *Unisal* a *SpectralResidual* sme použili metriky AUC-Judd a NSS.

Tabuľka 5.5: Vyhodnotenie metód detekcie významnosti.

Metóda	AUC-Judd	NSS
<i>UniSal</i>	0,926	2,287
<i>SpectralResidual</i>	0,915	2,921

Ideálna hodnota metriky AUC-Judd je 1, čo znamená, že ide o 100% úspešnosť. Náhodná mapa významností má výslednú hodnotu 0,5, čo znamená, že pomocou našich metód by sme mali získať hodnotu väčšiu ako 0,5. V tabuľke 5.6 môžeme vidieť, že obe použité metódy obstáli výborne a na testovacom datasete dosiahli slušné výsledky.

5.2.3 Detekcia významnosti reklamnej plochy

Pri klasifikácii významnosti reklamných plôch mohli nastať tieto prípady:

- True positive (TP) - reklamná plocha bola správne označená ako významná, keď sa na nej nachádzala fixácia očí,
- True negative (TN) - reklamná plocha správne nebola označená ako významná, keď sa na nej nenachádzala fixácia očí,
- False positive (FP) - reklamná plocha bola označená ako významná, keď sa na nej nenachádzala fixácia očí,

- False negative (FN) - reklamná plocha nebola označená ako významná, keď sa na nej nachádzala fixácia očí.

Dôležitým krokom bolo získanie priemernej hodnoty významnosti všetkých reklamných plôch na tréningovej množine. Priemerná hodnota významnosti reklamných plôch na tréningovej množine, ktorá bola normalizovaná na intervale od 0 do 1 bola 0,416.

Tabuľka 5.6: Vyhodnotenie metód detekcie významnosti reklamných plôch.

Metóda	TP	TN	FP	FN	Precision	Recall
<i>UniSal</i>	114	32	24	40	82,6%	74,0%
<i>SpectralResidual</i>	108	24	48	30	69,2%	78,3%

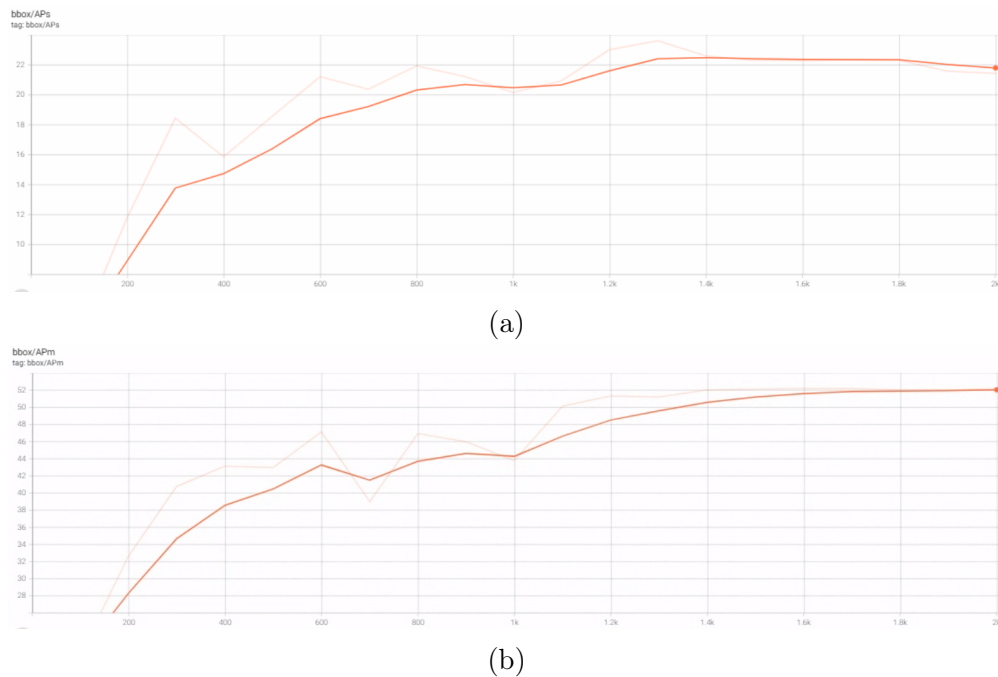
V tabuľke 5.6 môžeme vidieť, že naša metóda na detekciu významnosti reklamných plôch dosiahla pri tvorbe mapy významností pomocou metódy *UniSal* presnosť 82,6% a citlivosť 74,0%. S použitím metódy *SpectralResidual* bola presnosť 69,2% a citlivosť 78,3%, čo znamená, že pri použití metódy *UniSal* je vyššia pravdepodobnosť, že predikované pozitívne detekcie boli naozaj správne.

5.3 Zhrnutie výsledkov

Z celkového vyhodnotenia vyplýva, že v úlohe detekcie reklamných plôch bol presnejší aj rýchlejší model YOLOv5. Najlepšie výsledky sa nám podarilo pri oboch modeloch dosiahnuť po dotrénovaní na vlastnom datasete. Dôvody horších výsledkov po dotrénovaní na Mapillary Vistas datasete sme uviedli v kapitole 5.1.3.

Na obrázku 5.5 v grafoch si môžeme všimnúť, že model Faster R-CNN mal aj po dotrénovaní problém s malými a strednými objektami.

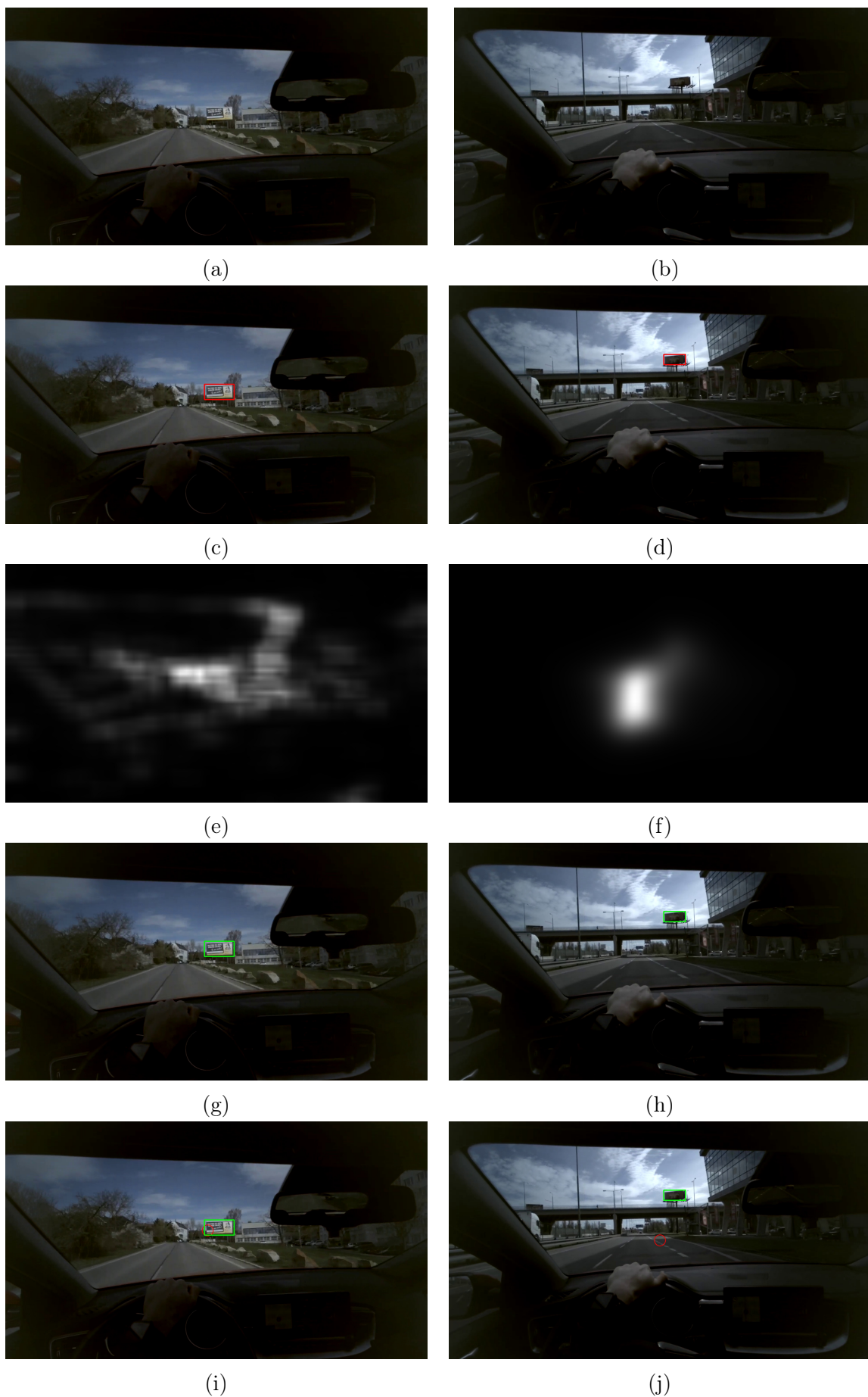
Pri riešení problému detekcie významností reklamných plôch sme najskôr otestovali metódy *UniSal* a *SpectralResidual*, pomocou ktorých sme pre každý obrázok vytvorili mapy významností. Pri vyhodnocovaní sme ako ground truth (skutočná informácia) použili mapu fixácií získanej z eyetrackera. Obe použité metódy na detekciu významných oblastí v obraze dosiahli slušné výsledky. Pri metóde *UniSal* by sa výsledky mohli vylepšiť, ak by tento model bol natrénovaný alebo dotrénovaný na datasete, ktorý obsahuje snímky alebo videá zosnímané pomocou eyetrackera vnútri vo vozidle. To by znamenalo, že by sa vedel lepšie prispôbiť rôznym situáciám cestnej premávky.



Obr. 5.5: Faster R-CNN - problém s malými a strednými objektami.

Na záver sme otestovali úspešnosť nami implementovanej metódy na detekciu významnosti reklamných plôch. Pri vyhodnotení sme prišli na to, že naša metóda dosahuje lepšie výsledky na mapách významností vytvorených pomocou metódy *UniSal*.

Žiaden algoritmus nedokáže dostatočne presne napodobniť správanie človeka ani ľudský mozog, preto pri detekcii významnosti reklamných plôch mohli nastať rôzne prípady chybnej detekcie. Jedným z týchto prípadov môže byť zvýšená pozornosť pri riadení automobilu, napríklad pri predbiehaní alebo iných udalostiach cestnej premávky. Ďalším problémom môže byť zlá detekcia reklamnej plochy, keď je za reklamnú plochu označená dopravná značka, prípadne farebná fasáda budovy.



Obr. 5.6: Ukážka správnej a nesprávnej detekcie významnosti reklamných plôch: a) a b) - Pôvodné obrázky, c) a d) - Detekcia reklamných plôch, e) a f) - Mapy významností, g) a h) - Detekcia významnosti reklamnej plochy, i) a j) - Verifikácia (červený kruh - fixácia z eyetrackera).

Záver

Práca sa zaoberá možnosťami využitia neurónových sietí na detekciu reklamných plôch pri cestách s cieľom zistiť relevantnosť a sledovanosť reklamných plôch popri cestách. V úvodnej časti je uvedený všeobecný prehľad problematiky reklamných plôch, detekcie objektov a popis podobných prác.

Práca je ďalej zameraná na popis neurónových sietí a významných oblastí. Uvedené sú základné charakteristiky neurónovej siete, porovnanie biologického neurónu s umeľým, vrátane popisu štruktúry neurónovej siete. V práci sú popísané základné princípy učenia neurónových sietí a jednotlivé funkcie podieľajúce sa na tréňovaní neurónovej siete.

Ďalšia časť práce je zameraná na popis detekcie objektov pomocou neurónových sietí a jednotlivých metód založených na hlbokom učení. Sú tu tiež popísané základné princípy fungovania jednotlivých metód, ich výhody a nevýhody a používané datasety. Ďalej je tu charakterizovaná detekcia významných oblastí v obraze, metódy a modely používané na detekciu významných oblastí.

Experimentálna časť práce bola zameraná na návrh a implementáciu metódy na detekciu reklamných plôch a významnosti reklamných plôch. V práci boli implementované dva modely na detekciu objektov, a to YOLOv5 a Faster R-CNN. Výkonnosť týchto modelov sa posudzovala na základe metriky Average Precision, ktorá sa používa pri súťažiach PASCAL VOC a COCO. Pri tréňovaní neurónových sietí platí, že sa ťažko konfigurujú a veľmi záleží na zvolení správnych vstupných parametrov. Ďalšia časť práce bola venovaná detekcii významnosti reklamných plôch, kde za účelom vytvorenia mapy významnosti bola použitá metóda, ktorá používa spektrálne reziduá a model UniSal, ktorý využíva princíp hlbokého učenia. Potom bola implementovaná metóda na detekciu významnosti reklamnej plochy, ktorá je založená na výpočte priemernej hodnoty významnosti v oblasti reklamnej plochy. Navrhnuté a implementované

metódy boli na záver otestované a vyhodnotené.

Jedným z vylepšení by mohlo byť vytvorenie modelu, ktorý by detekciu reklamnej plochy a jej významnosti vykonával naraz v jednom kroku a aplikácia by fungovala v reálnom čase. Aplikácia by ukladala informácie o reklamných plochách, na základe ktorých by sa dalo určiť, ktoré druhy reklamných plôch sú najviac významné, čo by bola dobrá spätná väzba pre marketing.

Literatúra

- [1] 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.
- [2] 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.
- [3] 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.
- [4] Kristína Miklošová. Výskum modelov významných oblastí malieb. Diplomová práca, 2020. Univerzita Komenského v Bratislave, Fakulta Matematiky, Fyziky a Informatiky.
- [5] David G. T. Barrett, Ari S. Morcos, and Jakob H. Macke. Analyzing biological and artificial neural networks: challenges with opportunities for synergy?, 2018.
- [6] Simon Haykin. *Neural Networks and Learning Machines*. Prentice Hall, 2008. Third Edition.
- [7] Framework torch. Dostupné na internete: <https://www.root.cz/clanky/framework-torch-vyuziti-konvolucnich-siti-pro-rozpoznavani-a-klasifikaci-obrazku/#k0>. Navštívené: 2022-4-7.
- [8] Basic overview of convolutional neural network. Dostupné na inter-

- nete: <https://medium.com/dataseries/basic-overview-of-convolutional-neural-network-cnn-4fcc7dbb4f17>.
- [9] What are hyperparameters? <https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a>. Navštívené: 2021-12-8.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] Ross Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context.

- [18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [19] Lisa dataset. Dostupné na internete: https://git-disl.github.io/GTDLBench/datasets/lisa_traffic_sign_dataset/.
- [20] Elena Šikudová, Zuzana Černeková, Wanda Benešová, Zuzana Haladová, and Júlia Kučerová. *Počítačové videnie Detekcia a rozpoznávanie objektov*. Vydavateľstvo Wikina, 2011.
- [21] Understanding eye tracking how it can work for you: Definitions, metrics, and applications. Dostupné na internete: <https://eyeware.tech/blog/what-is-eye-tracking/>.
- [22] Amer Al-Rahayfeh and Miad Faezipour. Eye tracking and head movement detection: A state-of-art survey. *IEEE Journal of Translational Engineering in Health and Medicine*, 1:2100212 –2100212, 2013.
- [23] Vidas Raudonis, Rimvydas Simutis, and Gintautas Narvydas. Discrete eye tracking for medical applications. In *2009 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies*, pages 1–6. IEEE, 2009.
- [24] Xiaohui Yang, Jiande Sun, Ju Liu, Jinyu Chu, Wei Liu, and Yuling Gao. A gaze tracking scheme for eye-based intelligent control. In *2010 8th World Congress on Intelligent Control and Automation*, pages 50–55. IEEE, 2010.
- [25] Yen-Wei Chen and Kenji Kubo. A robust eye detection and tracking technique using gabor filters. In *Proceedings of the Third International Conference on International Information Hiding and Multimedia Signal Processing (IIH-MSP 2007) - Volume 01*, IIH-MSP '07, page 109–112, USA, 2007. IEEE Computer Society.
- [26] Dong Hyun Yoo, Jae Heon Kim, Bang Rae Lee, and Myoung Jin Chung. Non-contact eye gaze tracking system by mapping of corneal reflections. In *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, pages 101–106, 2002.

- [27] Zhiwei Zhu and Qiang Ji. Novel eye gaze tracking techniques under natural head movement. *IEEE Transactions on Biomedical Engineering*, 54(12):2246–2260, 2007.
- [28] R.C. Coetzer and G.P. Hancke. Eye detection for a real-time vehicle driver fatigue monitoring system. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 66–71, 2011.
- [29] Tobii glasses 3. Dostupné na internete: <https://ifdesign.com/en/winner-ranking/project/tobii-pro-glasses-3/300945>.
- [30] 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.
- [31] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
- [32] Ming Jiang, Shengsheng Huang, Juanyong Duan, and Qi Zhao. Salicon: Saliency in context. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
- [33] Wenguan Wang, Jianbing Shen, Fang Guo, Ming-Ming Cheng, and Ali Borji. Revisiting video saliency: A large-scale benchmark and a new model. *CoRR*, abs/1801.07424, 2018.
- [34] 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.
- [35] Richard Droste, Jianbo Jiao, and J. Alison Noble. Unified Image and Video Saliency Modeling. In *Proceedings of the 16th European Conference on Computer Vision (ECCV)*, 2020.
- [36] Pytorch. Dostupné na internete: <https://pytorch.org/>.

- [37] Roboflow. Dostupné na internete: <https://roboflow.com/>.
- [38] YoloV5. Dostupné na internete: <https://github.com/ultralytics/yolov5>.
- [39] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. Dostupné na internete: <https://github.com/facebookresearch/detectron2>, 2019.
- [40] Github - unisal. Dostupné na internete: <https://github.com/rdroste/unisal>.
- [41] Nicolas Riche, Matthieu Duvinage, Matei Mancas, Bernard Gosselin, and Thierry Dutoit. Saliency and human fixations: State-of-the-art and study of comparison metrics. In *2013 IEEE International Conference on Computer Vision*, pages 1153–1160, 2013.
- [42] Reagan Galvez, Argel Bandala, Elmer Dadios, Ryan Vicerra, and Jose Martin Maningo. Object detection using convolutional neural networks. pages 2023–2027, 10 2018.
- [43] 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.
- [44] 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.
- [45] Sebastian Montabone and Álvaro Soto. Human detection using a mobile platform and novel features derived from a visual saliency mechanism. *Image Vis. Comput.*, 28:391–402, 2010.

Zoznam obrázkov

1.1	Architektúra siete AlexNet	3
1.2	Architektúra siete ADNet	4
1.3	Architektúra siete YOLO	5
1.4	Výsledok detekcie účastníkov cestnej premávky	5
1.5	Saliency Mapy z rôznych modelov	6
2.1	Schéma biologického neurónu	7
2.2	Schéma umelého neurónu	8
2.3	Ukážka konvolučnej neurónovej siete	9
2.4	Aktivačné funkcie	11
2.5	Príklad miery učenia	12
2.6	RCNN architektrúta	13
2.7	Fast RCNN architektrúta	14
2.8	Faster RCNN architektrúta	14
2.9	SSD architektrúta	15
2.10	YOLO architektrúta	15
2.11	Príklad LISA	16
2.12	Eyetracker	19
2.13	Ittiho model	20
2.14	Spektrálne reziduá	21
2.15	DHF1K Dataset	21
3.1	Návrh k postupu detekcie významnosti reklamnej plochy.	24
4.1	Tobii Glasses 3	26
4.2	Mapillary Vistas Dataset ukážky	27

4.3	Vlastný dataset ukážky	28
4.4	Ukážka anotovanie	28
4.5	Prekrytá reklamná plocha	29
4.6	Príklad textového súboru s anotáciami objektov	30
4.7	Predtrénované checkpointy YOLO	31
4.8	Predtrénované checkpointy Faster R-CNN	31
4.9	Ukážka grafov chybových funkcií	33
4.10	Architektúra UniSal	34
4.11	Vytváranie mapy významnosti	35
4.12	Pipeline - Detekcia významnosti na základe hodnôt významnosti.	36
5.1	Metrika IoU	39
5.2	Ukážka metriky IoU	39
5.3	Ukážka č.1 anotovaných malých objektov	42
5.4	Ukážka č.2 anotovaných malých objektov	42
5.5	Faster R-CNN - problém s malými a strednými objektami	47
5.6	Ukážka detekcie významnosti reklamných plôch	48