



Univerzitet u Novom Sadu Fakultet tehničkih nauka

Inženjerstvo informacionih sistema

Detekcija objekata

Sadržaj

1. Uvod u problem.....	3
1.1. Definicija detekcije objekata.....	3
1.2. Ključni izazovi kod detekcije vozila	3
2. Pregled literature i postojećih pristupa	3
2.1. Evolucija pristupa	3
2.2. Ključni naučni radovi	4
2.3. Skupovi podataka.....	4
3. Najčešće korišćeni algoritmi i metrike	4
3.1. Algoritmi detekcije (Arhitekture)	4
3.2. Metrike evaluacije.....	5
4. Komentar i ideje studenta	6
4.1. Analiza trenutnog stanja i mog zadatka	6
4.2. Vizija rešenja i plan rada	6
4.3. Šta očekujem kao najveći izazov?.....	7
5. Literatura.....	7

1. Uvod u problem

1.1. Definicija detekcije objekata

Detekcija objekata (**Object Detection**) predstavlja fundamentalni zadatak kompjuterske vizije koji kombinuje dva ključna aspekta: **lokalizaciju** objekata u slici (određivanje pozicije pomoću graničnih okvira – *bounding boxes*) i njihovu **klasifikaciju** (prepoznavanje kojoj klasi objekat pripada).

U kontekstu mog projekta, fokus je na detekciji vozila. Dok obična klasifikacija samo potvrđuje da se na slici nalazi automobil, detekcija nam omogućava da precizno lociramo svako vozilo na saobraćajnici, što je ključno za praćenje saobraćaja, brojanje vozila i analizu ponašanja učesnika u saobraćaju.

1.2. Ključni izazovi kod detekcije vozila

Detekcija automobila na saobraćajnicama suočava se sa specifičnim izazovima:

- **Varijabilnost razmere:** Automobili blizu kamere su veliki, dok su oni u daljini (na horizontu saobraćajnice) predstavljeni sa svega nekoliko piksela.
- **Okluzija:** Vozila se često delimično zaklanjaju međusobno ili ih zaklanjaju elementi infrastrukture (stubovi, drveće, saobraćajni znaci).
- **Ugao posmatranja:** Izgled automobila se drastično menja u zavisnosti od toga da li je kamera postavljena direktno iznad puta (bird's-eye view) ili pod uglom (standardni video nadzor).
- **Vremenski uslovi:** Senke, odsjaji na asfaltu, kiša ili noćni uslovi direktno utiču na vidljivost karakteristika vozila.

2. Pregled literature i postojećih pristupa

2.1. Evolucija pristupa

1. **Era tradicionalnih metoda (pre 2012):** Korišćenje ručno dizajniranih osobina poput **HOG** (*Histogram of Oriented Gradients*) za detekciju oblika vozila i **Haar kaskada** koje su se inicijalno koristile za lica, ali su adaptirane i za automobile.
2. **Region-based metode (2014–2016):** Revolucija sa **R-CNN** serijom[1]. **Faster R-CNN**[2] je uveo *Region Proposal Network* (RPN) koji omogućava mreži da sama "predloži" gde se na putu nalaze potencijalni automobili pre finalne klasifikacije.
3. **Era jednoetapnih detektora (2016–danas):** **YOLO** (*You Only Look Once*)[3] i **SSD** su omogućili detekciju u realnom vremenu, što je kritično za pametne saobraćajne sisteme. Ovi modeli obrađuju sliku u jednom prolazu, tretirajući detekciju kao problem regresije.

4. **Moderna era (2020–danas): DETR** (*Detection Transformer*) koristi mehanizam pažnje (attention) da modelira relacije između objekata na slici, što pomaže u razumevanju konteksta (npr. automobili se obično nalaze na traci, a ne na trotoaru).

2.2. Ključni naučni radovi

- **Faster R-CNN [2]:** Ovaj model je napravio revoluciju uvođenjem **Region Proposal Network (RPN)** modula, koji deli konvolucione slojeve sa mrežom za detekciju. Ovim je eliminisana potreba za eksternim i sporim algoritmima poput *Selective Search-a*, što je drastično ubrzalo proces generisanja predloga regiona. Postao je zlatni standard za dvo-fazne (two-stage) detektore jer je omogućio skoro real-time rad uz zadržavanje izuzetno visoke preciznosti. Njegova arhitektura je i danas temelj za mnoge savremene sisteme u medicinskoj dijagnostici i satelitskom snimanju.
- **YOLO serija [3,4]:** YOLO arhitektura pristupa detekciji objekata kao jedinstvenom **regresionom problemu**, gde se koordinate okvira i verovatnoće klasa predviđaju u samo jednom prolazu kroz mrežu. Za razliku od dvo-faznih modela, YOLO posmatra celu sliku odjednom, što mu omogućava da postigne ekstremno visoke frekvencije osvežavanja (45+ FPS), neophodne za autonomnu vožnju i video nadzor. Kroz svoje brojne iteracije (poput v4 i kasnijih verzija), model je optimizovan korišćenjem naprednih tehnika poput *Bag-of-Freebies* i *Bag-of-Specials* kako bi se zadržala brzina uz značajno povećanje tačnosti.
- **EfficientDet [5]:** Razvijen od strane Google Research tima, ovaj model uvodi inovativni **BiFPN (Bidirectional Feature Pyramid Network)** koji omogućava brzu i laku fuziju karakteristika sa različitih nivoa rezolucije u oba smera. Ključna inovacija je i metoda **složenog skaliranja (compound scaling)**, kojom se rezolucija slike, dubina i širina mreže podešavaju istovremeno na osnovu dostupnih resursa. Zahvaljujući ovim optimizacijama, EfficientDet postiže vrhunske rezultate (SOTA) uz 4 do 9 puta manji broj parametara i znatno manju računarsku zahtevnost (FLOPs) u odnosu na prethodne modele.

2.3. Skupovi podataka

Za detekciju vozila najznačajniji su:

- **MS COCO[6] :** Sadrži hiljade slika automobila, kamiona i motocikala u realnim scenama.
- **KITTI:** Specijalizovan za autonomnu vožnju, sa fokusom na gradski saobraćaj.
- **Cityscapes:** Fokusira se na semantičko razumevanje urbanih sredina.

3. Najčešće korišćeni algoritmi i metrike

3.1. Algoritmi detekcije (Arhitecture)

Algoritme generalno delimo u tri glavne grupe:

3.1.1. Dvoetaetni detektori (*Two-stage detectors*)

Ovi modeli prvo traže "zanimljive" regione na slici gde bi mogao biti auto, a zatim te regione klasifikuju.

- **Faster R-CNN**[2]: Najpoznatiji predstavnik. Koristi **RPN** (*Region Proposal Network*) koji skenira saobraćajnicu i predlaže koordinate gde se potencijalno nalaze vozila. Iako je izuzetno precizan (naročito za male automobile u daljini), sporiji je jer mora dva puta da "pogleda" delove slike.
- **Primena**: Koristi se kada nam brzina nije presudna, već maksimalna tačnost (npr. analiza snimaka za policijske uviđaje).

3.1.2. Jednoetaetni detektori (*One-stage detectors*)

Ovi modeli preskaču korak predlaganja regiona i direktno predviđaju klase i okvire u jednom prolazu kroz mrežu.

- **YOLO (You Only Look Once)** [3]: Trenutno najpopularniji za saobraćaj. Slika se deli na mrežu (grid), i svaka ćelija mreže je odgovorna za predviđanje vozila ako se centar vozila nalazi u njoj. **YOLOv8** i **YOLOv9** su idealni za moj rad jer su optimizovani da prepoznaju automobile u realnom vremenu.
- **SSD (Single Shot MultiBox Detector)**: Slično kao YOLO, ali koristi mape karakteristika različitih rezolucija kako bi bolje detektovao objekte različitih veličina (npr. autobus blizu kamere i auto u daljini).
- **Primena**: Idealno za pametne raskrsnice i autonomna vozila gde se odluka mora doneti u milisekundama.

3.1.3. Metode bez sidra (*Anchor-free methods*)

Noviji pristup koji ne koristi unapred definisane pravougaonike (anchors) već detektuje vozila kao ključne tačke (npr. centar auta).

- **CenterNet / FCOS**: Ovi modeli često bolje rešavaju problem preklapanja automobila na zakrčenim saobraćajnicama jer se fokusiraju na centar mase objekta.

3.2. Metrike evaluacije

Da bismo znali koliko je naš model dobar u prepoznavanju automobila, koristimo:

- **Intersection over Union (IoU)**: Mera preklapanja predviđenog i stvarnog okvira.

$\text{IoU} = \text{Area of Overlap} \setminus \text{Area of Union}$

- **mAP (mean Average Precision):** Glavna metrika koja balansira preciznost i odziv kroz sve klase vozila.
- **FPS (Frames Per Second):** Ključno ako planiramo da model koristimo za video nadzor saobraćaja uživo.

4. Komentar i ideje studenta

4.1. Analiza trenutnog stanja i mog zadatka

Nakon što sam pregledao literaturu i analizirao dataset koji sam dobio, shvatio sam da detekcija automobila na saobraćajnicama nije nimalo jednostavan zadatak, iako je oblast dobro istražena.

Za realizaciju praktičnog dela projekta korišću **skup podataka** koji sadrži ukupno 1176 slika relativno visokog kvaliteta, ekstrahovanih iz video zapisa u realnim uslovima saobraćaja. Dataset je podeljen na set za obučavanje od 1001 slike i testni set od 175 slika. Fokus detekcije je na klasi 'automobil', gde su svi objekti precizno anotirani putem graničnih okvira (bounding boxes). Anotacije su obezbeđene u CSV formatu i definisane su koordinatama $[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$, što će mi omogućiti mapiranje objekata na slici i pruža osnovu za treniranje modela zasnovanih na dubokom učenju.

Primećujem da su se ranije koristile komplikovane metode iz više koraka, ali za moj rad smatram da je **jednoetačni pristup (single-stage)** mnogo logičniji. Razlog je taj što saobraćajni podaci zahtevaju brzinu — nema svrhe detektovati auto ako je on već prošao kroz raskršnicu dok je model još uvek procesirao sliku.

4.2. Vizija rešenja i plan rada

Za svoj projektni zadatak odlučio sam da koristim **YOLOv8** arhitekturu. Izabrao sam je jer mi se čini da je trenutno najbolja u pogledu balansa između toga koliko je laka za programiranje i rezultata koje daje.

Moj plan je sledeći:

1. **Priprema podataka:** Dobijene slike saobraćajnice ću prvo pregledati i očistiti. Ako primetim da imam malo slika u noćnim uslovima, korišću **data augmentation** (veštačko menjanje osvetljenja, rotacije, "mosaic" spajanje slika) kako bih "naterao" model da nauči da prepoznaje automobile i u težim uslovima.
2. **Fine-tuning:** Neću trenirati model od nule (jer nemam hiljade slika i procesorsku snagu za to), već ću uzeti model koji je već "video" milione slika (pre-trained na COCO datasetu) i prilagoditi ga specifično mojim slikama saobraćajnice.

3. **Rešavanje problema malih objekata:** Planiram da eksperimentišem sa rezolucijom ulaza. Pošto su automobili u daljini sitni, veća rezolucija ulazne slike bi mogla pomoći modelu da ih "uoči", ali moram paziti da to ne uspori previše rad samog programa.

4.3. Šta očekujem kao najveći izazov?

Najveći izazov predstavljaće **okluzija** — kada jedan kamion zakloni dva automobila iza sebe. Tu model često pogreši i ne detektuje sakrivena vozila. Takođe, očekujem da će biti teško razlikovati slične klase, poput kombija i manjih kamiona, na slikama niske rezolucije i to što su slike auta u pokretu pa se na nekim slikama gube jasne ivice automobila. Moje rešenje će biti pažljivo podešavanje **IoU praga** i **confidence threshold-a** kako bih smanjio broj lažno pozitivnih detekcija.

5. Literatura

- [1] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. <https://doi.org/10.1109/CVPR.2014.81> IEEE Xplore
- [2] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 28, 91–99. <https://doi.org/10.1109/TPAMI.2016.2577031> IEEE Xplore
- [3] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. <https://doi.org/10.1109/CVPR.2016.91> IEEE Xplore
- [4] Bochkovski, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint*. <https://doi.org/10.48550/arXiv.2004.10934> arXiv
- [5] Tan, M., Pang, R., & Le, Q. V. (2020). EfficientDet: Scalable and Efficient Object Detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10781–10790. <https://doi.org/10.1109/CVPR42600.2020.01080> IEEE Xplore
- [6] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*. https://doi.org/10.1007/978-3-319-10602-1_48 Springer Link