

Prepoznavanje objektov na trgovinskih policah za robotske aplikacije

Blaž Potočnik, Matjaž Mihelj, Marko Munih, Janez Podobnik

Fakulteta za elektrotehniko, Univerza v Ljubljani, Tržaška c. 25, 1000 Ljubljana

E-pošta: bp1376@student.uni-lj.si

Object recognition on shelves for robotics applications

Abstract. In this paper, we focus on preparing solutions for recognition and detection of objects in densely packed scenes such as store shelves. Goal is to be able to use prepared solutions for object detection based on neural networks for automation in retail for online orders. We focused on object recognition and detection, furthermore we also investigated product recognition and recognition of empty spaces on shelves. Main goal was to test how pertained models will work on photos from local store.

1 Uvod

Prepoznavanje objektov, ljudi, vzorcev na slikah ni nič novega. Vse to se že dolgo uporablja v realnem svetu. Lahko govorimo o uporabi v policijske namene, to so prepoznavanje registrskih tablic, lahko pa imamo v mislih le preprosto odklepanje telefona s srednjo kamero, ki prepozna značilnice vašega obraza. Eden od potencialnih problemov je tudi prepoznavanje trgovinskih objektov na policah. To bi omogočilo avtomatizirano določanje pravilne razporeditve artiklov na policah, določanje mest s praznimi policami, in tudi robotizirano pobiranje artiklov s polic [9].

Vendar se izkaže, da se na tem področju pojavljajo specifični problemi in pogoji. Prvi problem, ki se pojavi je velika gostota izdelkov na policah [10]. Drugi problem pa je velika razlika med izgledom embalaže izdelkov na eni strani in veliko primerov iste embalaže na istih slikah. Če želimo zaznati objekt, moramo zaznati tako majhen paket žvečil, kot tudi največje pakiranje pralnega praška, različne velikosti platenk in tako naprej. Čeprav nevronske mreže niso vedno odgovor na vsako vprašanje, kot se jih v zadnjem času rado predstavlja, je tukaj njihova uporaba povsem upravičena. Že takoj na začetku, smo se torej odločili, da bo projekt temeljil na uporabi nevronske mreže, odločili smo se za nevronske mreže, ki je naučena na bazi slik SKU-110K [1].

Cilj naloge je bil pripraviti zaznavanje objektov na trgovinskih policah. Razlog, za zastavitev take naloge je želja podjetja Interspar, da bi bilo skladišče za njihovo spletno trgovino popolnoma avtomatizirano. Prepoznavanje objektov in detekcija njihovega položaja bi omogočilo razvoj robotskih aplikacij za pobiranje trgovinskih artiklov iz polic.

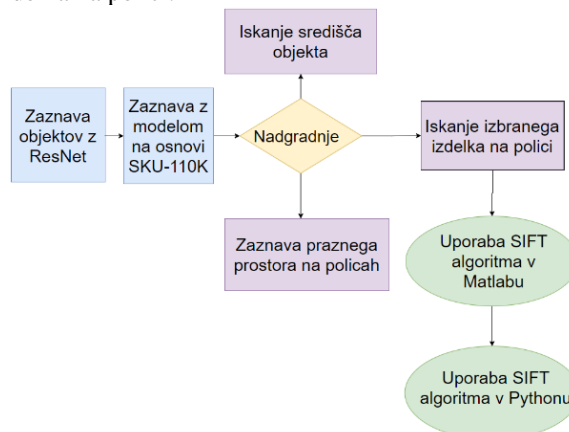
2 Koncept avtomatizirane izvedbe naročila

Zamišljeni proces avtomatizirane izvedbe naročila je sledeč: prvi korak je, da kupec odda naročilo v spletni trgovini. To naročilo se na strežniku trgovine obdela, poleg podatkov, kdo je naročnik, kakšen je naslov za

dostavo in tako naprej bi se v tem primeru iz podatkovne baze za vsak naročen artikel prebrala še lokacija artikla (oznaka regala), slika artikla in dimenzije artikla. Glede na naročene artikle pa bi se generirala tudi optimalna pot med naročenimi izdelki. Ti podatki se posredujejo robotu, ki se odpravi izdelke iskat v skladišče. Potuje po predpisani, ko zazna, da je pred prvim markerjem, se ustavi in zajame planogram (planogram je fotografija regala). Iz te fotografije regala najprej pridobi objekte in njihova središča. Potem vsak objekt primerja s posredovano sliko ciljnega artikla. Ko zazna ujemanje, se na podlagi središča artikla in njegovih dimenzij pripravi za prijemanje, vodenje do artikla poteka na podlagi robotskega vida. Artikel pobere in ga spravi v »košarico«. To ponovi za vse artikle, ki jih naročilo vsebuje. Ko pobere vse artikle, se vrne na predpisano lokacijo, kamor odloži nakup. Ta nakup kurir le še pobere in ga dostavi naročniku. Iz te fotografije regala najprej pridobi objekte in njihova središča. Potem vsak objekt primerja s posredovano sliko ciljnega artikla. Ko zazna ujemanje, se na podlagi središča artikla in njegovih dimenzij pripravi za prijemanje, vodenje do artikla poteka na podlagi robotskega vida. Artikel pobere in ga spravi v »košarico«. To ponovi za vse artikle, ki jih naročilo vsebuje. Ko pobere vse artikle, se vrne na predpisano lokacijo, kamor odloži nakup. Ta nakup kurir le še pobere in ga dostavi naročniku.

3 Izvedba

Sedaj pa si pogledajmo še, katere potrebne elemente smo že uspeli pripraviti in čemu se bo v prihodnosti potrebno še posvetiti. Na sliki 1 je prikazan diagram poteka našega dela. Z modro barvo je označena naša osnovna naloga, zaznave objektov na policah, z vijolično so označene 3 nadgradnje prepoznavanja objektov katerih smo se lotili. Z zeleno pa je prikazan potek dela pri iskanju izbranega izdelka na polici.



Slika 1. Diagram poteka prepoznavanja in detekcije objektov

3.1 Zaznavanje objektov na slikah

Naše delo smo začeli s pripravo programskega okolja in zagonom že pripravljene kode, ki je bila prikazana na konferenci CVPR2019 [1]. Nevronska mreža SKU-110K je bila zgrajena na osnovi zelo znanega detektorja RetinaNet [2]. Ker smo se prvič ukvarjali s strojnim učenjem, smo se odločili, da najprej poizkusimo zagnati model RetinaNet, saj je bil bolj preprost. Za razliko od modela SKU-110K, RetinaNet vsebuje več kategorij, torej razlikuje med objekti. Ni pa namenjen zaznavanju objektov, ki so blizu skupaj. Po uspešno postavljenem programskem okolju na lokalnem računalniku, je nevronska mreža RetinaNet vrnila rezultate, ki jih lahko vidite na sliki 2.



Slika 2. Rezultat delovanja RetinaNet algoritma.

Rezultat prepoznave ni dosegal željene kvalitete. Zaznalo je le nekaj plastenk, tiste, ki jih je zaznalo, pa je zaznalo več kot enkrat. Ta večkratna zaznava istih plastenk oziroma objektov v splošnem, je glavna značilnost modela, ki bazira na RetinaNet-u.

Naslednji korak je bil preizkus izbranega primera zaznavanja objektov na trgovinskih policah. Ta temelji na bazi slik SKU-110K [1]. Ker je učenje dolgotrajno, smo uporabili že naučen model. Glavna značilnost tega modela je, da preprečuje večkratno zaznavo istih izdelkov in da vsebuje le kategorijo »objekt«, ni pa sposoben med objekti razlikovati. Ta del smo izvedli na platformi Microsoft Azure. Več o platformi in pripravi platforme za delo si je mogoče prebrati v dodatnih navodilih [3].

Ko smo ugotovili, da deluje na slikah, ki so jih razvijalci modela uporabili za učenje in testiranje, je bil naslednji korak preizkus na lastnih slikah. Za prvi test smo posneli nekaj slik produktov na policah v Laboratoriju za robotiko. Police je prispevalo podjetje Spar in so take, kot se uporabljajo v trgovinah. Na njej so zloženi izdelki, vendar v tem primeru ne tako gosto skupaj kot v trgovini. Rezultati so bili obetavni in jih lahko vidite na sliki 3. Detektor je zaznal večino objektov, bilo je tudi nekaj napačnih detekcij. Opazimo, da je napačno označilo robota, ki se nahaja v ozadju pa tudi stol. Na splošno slika 3 ne predstavlja idealne situacije za izbrani model, saj je na njej preveč okolja, ki moti zaznavo.



Slika 3. Delovanje modela na osnovi SKU 110-K na sliki police v laboratoriju.

Da bi lahko model preizkusili še na bolj resničnih situacijah, smo potrebovali še slike iz trgovine Spar. Sodelavci iz Laboratorija za robotiko so posneli 47 fotografij v trgovini Interspar v Cityparku. Te slike so naša testna množica. Test zaznave objektov na teh slikah je trajal približno 1 uro. Iskanje objektov na posamezni sliki je trajalo od 30 s pa do 90 s. Veliko je bilo odvisno od gostote izdelkov na sliki pa tudi same količine izdelkov. Primeri rezultatov so na slikah 4, 5 in 6.



Slika 4. Delovanje modela na sliki iz resnične trgovine.

Vse slike 4, 5 in 6 so izrez iz celotne slike, ki je bila v obdelavi, da se lažje videti podrobnosti za oceno kako algoritem deluje. Na sliki 4 je zaznava mleka. Mleko, kartonske kvadratne škatle, in podobno so zelo primerni objekti za zaznavo s tem modelom, saj so pravilnih oblik, naloženi eden poleg drugega, ne pa naključno. Zato je tudi delovanje ustrezno.



Slika 5. Zaznavanje artiklov in prikaz zaznave odsefov od polic.

Slika 5 prikazuje polico s škatlami jajc. Sama zaznava deluje dobro, problematični pa so odsevi škatel na polici. Polica je svetleča in zaradi močne osvetlitve so odsevi precej močni, zato jih model, ki je občutljiv prepozna. Ena od možnih rešitev bi bila zato v končni aplikaciji uporaba bolj matiranih površin polic, s čimer se izognemo problematičnim odbojem.

Na sliki 6 so šamponi in drugi izdelki za osebno nego. Taki izdelki pa so za model že težji test, saj niso nujno vsi lepo obrnjeni, na primer Elseve, izdelek na desni strani slike.



Slika 6. Problem lažne zaznave etiket s ceno.

Zanimivo pa modelu SKU-110K v tem primeru ni predstavljalo težave prozorno varovalo, ki se nahaja na polici in skrbi, da izdelki ne padajo iz police. Ena od napačnih zaznav pa so velike ploščice z akcijskimi cenami, ki jih uporablja Spar za označevanje. Očitno jih trgovine, kjer je bilo posnetih 10.000 slik, ki so se uporabljale za učenje, niso uporabljale, vsaj ne pogosto, zato model ni pravilno naučen za take primere. Dejstvo pa je, da tega problema pri uporabi za katero je namenjen ta algoritem, torej v trgovini brez fizičnih kupcev ne bo, saj tam ni potrebe za tako velike oznake.

3.2 Določanje središče izdelka

Zaznavanje objektov na trgovinskih policah je bil osnoven cilj, ki smo si jo zadali. Naslednji korak je bil izračun središča zaznanega objekta, saj je to skupaj z dimenzijami objekta podatek, ki omogoča pobiranje objekta s prijemalom. Za to je bilo potrebno spreminjanje datoteke Visualization.py. V tej datoteki je koda za izrisovanje pravokotnikov okrog zaznanih objektov. Ker lahko iz podatkov, ki se uporabljajo za izris pravokotnikov izračunamo tudi njihovo središče smo morali le dopisati dve funkciji v katerih se središče izračuna in potem tudi izriše. Rezultat je viden na sliki 7, središče izdelka predstavlja rdeča pika.



Slika 7. Zaznava središča objekta.

3.3 Prepoznavanje izbranega izdelka

Naslednja nadgradnja je bila izvedbe prepoznavanja specifičnih objektov. Kot že omenjeno nevrnska mreža zaznava objekte na policah, vendar med njimi ne razlikuje. Klasifikacija vrne samo en razred »objekt«. Škatlo pralnega praška, mleko, plastenko soka in vrečko testenin dojema enako, kot »objekt«. Odločili smo se uporabiti algoritem, ki išče korespondenčne točke in je neodvisen od skaliranja. Bolj pogosto ime za to je SIFT algoritem [4].

Cilj, ki smo si ga zastavili v tem delu je, da podamo sliko zelenega izdelka, v realnosti bi bil to izdelek, ki ga je stranka naročila. Glede na sliko zelenega izdelka, ki smo jo pridobili na Sparovi spletni trgovini, bi med slikami izdelkov izrezanimi iz planograma moral poiskati pravi izdelek.

Za začetek sem program poizkusil napisati v jeziku Matlab [5]. V programu sem definiriral objekt, ki ga iščemo in pa objekte, med katerimi iščemo. Za vsako sliko posebej smo poiskali korespondenčne pare točk med sliko iskanega objekta in sliko objekta iz trgovske police. Nato smo glede na število korespondenčnih parov določili, kateri od objektov je pravi.

Ker pa je vsa predhodna koda napisana v jeziku Python sem se odločil algoritem prepoznavanja SIFT [6] prestaviti v Python in ga še izboljšati. Ideja in izvedba sta bili enaki, glavni spremembi sta bili avtomatsko spreminjanje velikosti vhodnih slik, za lažjo primerjavo s sliko podanega izdelka. Avtomatizirali pa smo tudi nalaganje slik, na katerih išče željeni izdelek [7]. V Matlab-u je bila vsaka slika naložena ročno, v izvedbi prepoznavanja izdelkov v Python-u pa je bilo potrebno le podati mapo s slikami in program je izmed vseh slik našel najboljše ujemanje avtomatsko.

Sam algoritem sicer ni še blizu končnega, problematična je na primer situacija, da je na polici, kjer bi robot iskal izdelek zmanjkalo naročenih izdelkov. V trenutnem stanju nas robot ne bi obvestil, da izdelka ni na voljo, ampak bi nam prinesel izdelek, ki je najbolj podoben naročenemu. Rešitev je morda zadnja nadgradnja, ki smo jo izvedli.



Slika 8. Prepoznavanje izbranega izdelka.

3.4 Iskanje praznih prostorov na polici

Kot dodatek pa smo poizkusili rešiti še problem, ki je morda bolj prisoten v trgovinah trenutno kot pa bi bil problem v robotizirani trgovini. Govorimo o praznih prostorih na policah, ki nastanejo, ko zmanjka določenih izdelkov. Načinov reševanja tega problema je več, tudi nevronske mreže. Mi smo se zaradi pomanjkanja časa odločili za preprost pristop z uporabo tehnik obdelave slik v Python-u.

Ideja je bila, da prazne prostore zaznavamo z zaznavanjem temnih delov slike [8]. Sliko najprej spremenimo iz barvne v črno-belo, poiščemo temne lise in okrog njih narišemo pravokotnik. Ker to samo po sebi ni delovalo, smo omejili maksimalne in minimalne velikosti pravokotnikov in potem izračunali povprečno vrednost slikovnih točk znotraj pravokotnika. S tem smo dodali še eno raven upravljanja. Rezultat je viden na sliki 9, z rdečimi pravokotniki so označeni zaznani izdelki, z zelenim pa prazen prostor.



Slika 9. Zaznavanje praznega prostora na polici.

4 Zaključek

Pri izdelavi projekta smo se dotaknili veliko različnih vsebin: vzpostavitev ustreznega programskega okolja, prepoznavanje lege objektov z nevronske mreže, obdelava slik, ter prepoznavanje zelo različnih artiklov. Uporabljali smo različna napredna orodja, ki so sploh omogočila izdelavo projekta, kot je Microsoftovo računalništvo v oblaku na platformi Azure.

Delovanje same nevronske mreže ni popolnoma brez težav, je pa ravno pred kratkim prišla na Github še izboljšana verzija [11]. To, da prihajajo izboljšane različice in da se na projektu še vedno dela vliva precej optimizma, da se bo razvoj nadaljeval tudi v prihodnje. Prepoznavanje izbranega izdelka in pa iskanje praznih prostorov delujeta, bi pa za uporabo v resni aplikaciji potrebovala dodaten razvoj in precej nadgradenj. Uporabo algoritma za prepoznavanje izdelka omejuje predvsem nezmožnost izreza posameznih slik označenih

objektov iz planograma. Uporaba algoritma za zaznavanje praznih prostorov pa je omejena na črne police, obstaja pa tudi nevarnost lažnih zaznav praznih prostorov nad izdelki na primer.

Nič ni narobe, če predpostavimo, da smo s tem projektom postavili smernice, za prihodnje delo na tem projektu. Nekateri deli projekta so razviti bolj, drugi manj ampak rezultati, ki smo jih pridobili do sedaj nakazujejo, da je potencial celotnega projekta velik.

5 Literatura:

- [1] E. Goldman, R. Herzig, A. Eisenschat, J. Goldberger and T. Hassner, "Precise Detection in Densely Packed Scenes," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5222-5231.
- [2] T. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal Loss for Dense Object Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 2, pp. 318-327, 2020.
- [3] Blaž Potočnik, *Azure-manual-FE-LJ*. 2021.
- [4] D. G. Lowe, "Object recognition from local scale-invariant features," Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999, pp. 1150-1157 vol.2
- [5] Object Detection in a Cluttered Scene Using Point Feature Matching - MATLAB & Simulink. <https://www.mathworks.com/help/vision/ug/object-detection-in-a-cluttered-scene-using-point-feature-matching.html> (pridobljeno jan. 16, 2021).
- [6] Introduction to SIFT (Scale-Invariant Feature Transform) — OpenCV-Python Tutorials; https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature_2d/py_sift_intro/py_sift_intro.html (pridobljeno jan. 16, 2021).
- [7] SIFT | How To Use SIFT For Image Matching In Python. <https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/> (pridobljeno jan. 16, 2021).
- [8] Identify black dots on lemon, *Stack Overflow*. <https://stackoverflow.com/questions/52279916/identify-black-dots-on-lemon> (pridobljeno jul. 27, 2021).
- [9] Wei Y, Tran S, Xu S, Kang B, Springer M. Deep Learning for Retail Product Recognition: Challenges and Techniques. *Comput Intell Neurosci*. 2020;2020:8875910.
- [10] Varadarajan S., Kant S., Srivastava M. M. Benchmark for generic product detection: a strong baseline for dense object detection. 2019. <https://arxiv.org/abs/1912.09476>. (pridobljeno jul. 27, 2021).
- [11] Tianze Rong, Yanjia Zhu, Hongxiang Cai, Yichao Xiong, A Solution to Product detection in Densely Packed Scenes, 2020, <https://arxiv.org/abs/2007.11946> (pridobljeno jul. 27, 2021).