

Skladiščni sistem za verifikacijo predmetov s pametnimi očali

Janez Križaj, Janez Perš in Vitomir Štruc

Fakulteta za elektrotehniko, Univerza v Ljubljani

E-pošta: janez.krizaj@fe.uni-lj.si

Warehouse system for object verification with smart glasses

This paper presents a warehouse product verification system that helps warehouse staff in performing their work. The proposed system sequentially captures the pictures of the objects that the warehouse worker encounters in his work and verifies if the object in the picture is the one that the worker is supposed to find in the warehouse. The system uses Android-powered smart glasses to capture image data, while the computationally intensive verification tasks are carried out on the server computer and are implemented using recent deep learning techniques. By doing so, the system is able to process images in near real-time and achieves high verification accuracy as is demonstrated by the performed experiments.

1 Uvod

V članku predstavimo podporni sistem, ki skladiščnim delavcem omogoča zanesljivejšo in po možnosti hitrejšo izvajanje svojih nalog. Pri obstoječem načinu dela v skladiščih delavec v informacijskem sistemu najprej poišče lokacijo določenega produkta v skladišču, na dotični lokaciji pa nato z ustreznim čitalnikom preveri ali gre res za iskani produkt. Razviti podporni sistem pa s pomočjo pametnih očal zajema slike objektov, s katerimi se srečuje skladiščni delavec pri svojem delu, in preveri ali gre za pravi objekt, t.j. objekt, ki naj bi ga delavec v skladišču poiskal. Z načrtovano tehnologijo je torej v postopek vpeljan avtomatiziran proces verifikacije produktov, ki je izveden z računalniško-podprto tehnologijo razpoznavanja objektov iz slikovnih podatkov, zajetih s pametnimi očali.

Glavni izziv in razvojni problem, s katerim smo se srečali pri udejanjenju podpornega sistema je problem verifikacije objektov na podlagi slikovnih podatkov, zajetih s pametnimi očali. Procesno verigo verifikacijskega postopka smo sestavili iz klasičnih gradnikov, kot je detektor objektov, modul za izračun značilik in modul za razvrščanje, pri čemer smo za detekcijo objektov in izračun značilik uporabili sodobne postopke, ki temeljijo na uporabi globokih nevronske mreže.

Zaradi relativno visoke računske zahtevnosti teh postopkov smo večji del procesne verige prenesli na strežnik z grafično kartico, ki podpira hitro izvajanje in učenje globokih mrež, pametna očala Vuzix, na katerih

teče operacijski sistem Android, pa nam služijo le za zajem in prenos slike na strežnik ter prikaz rezultata verifikacije. Ob predpostavki dovolj hitre internetne povezave smo na ta način zagotovili sprejem povratne informacije, v obliki rezultata verifikacije in koordinat detekcijskega okna, v okvirno 0,2 s po tem, ko smo poslali sliko v obdelavo na strežnik.

2 Implementacija podpornega sistema

To poglavje vsebuje opis gradnikov in delovanje podpornega sistema za skladiščne delavce.

2.1 Način delovanja

Demonstracijski sistem za predmet, ki ga demonstrator prime v roke izvede verifikacijo. Sistem prične s poskusi razpoznavanja predmetov na zajetih slikah, ko skladiščnik z izbiro zelenega objekta na skladiščnem računalniku sproži postopek verifikacije. Od začetka razpoznavanja se z določenim intervalom vzorčenja zajemajo slike in pošljejo v obdelavo na strežnik. V primeru, da je istovetnost predmeta potrjena, se sistem ne odzove, v primeru, da sistemu istovetnosti predmeta ne uspe potrditi, pa sistem sproži "alarm" in uporabnika opozori, da trenutno aktualen predmet, ne ustreza predmetu, ki naj bi ga skladiščnik poiskal. Ob vsakem "alarmu" demonstracijski sistem shrani tudi sliko zajetega predmeta za namene arhiviranja.

2.2 Zgradba sistema

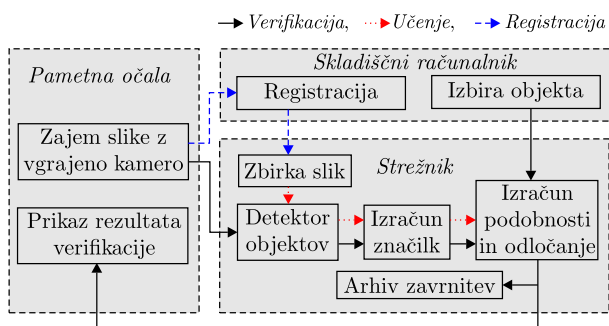
Kot je prikazano na sliki 1 je podporni sistem v grobem sestavljen iz treh delov:

- *Pametna očala:* Na pametnih očalih teče spletna aplikacija, ki omogoča zajem slike z vgrajeno kamero očal, prikaz slike na zaslonu očal, pošiljanje slike v obdelavo na strežnik v formatu JPEG, sprejem rezultata verifikacije iz strežnika in prikaz rezultata vključno z detekcijskim oknom na zaslonu očal;
- *Strežnik:* Ob sprejemu slike, strežnik pošlje dano sliko skozi procesno verigo, zgrajeno iz klasičnih gradnikov razpoznavnega sistema, kot so modul za predobdelavo slik, modul za izračun značilik, modul za merjenje podobnosti in odločanje. Strežnik nato izračunan rezultat verifikacije in koordinate detekcijskega okna za vsako dobljeno sliko pošlje nazaj do pametnih očal in

obenem arhivira slike pri katerih je prišlo do zavrnitve pri verifikaciji;

- *Skladišni računalnik* na katerem se izvaja registracija novih objektov v sistem in izbira objekta, ki ga želimo verifilirati.

Sledeča podglavja vsebujejo podrobnejši opis naštetih sklopov razvitega sistema.



Slika 1: Shematski prikaz podpornega sistema.

2.3 Registracija novih objektov v sistem

Pri registraciji objekta v sistem, se predpostavlja, da imamo na voljo 6 slik vsake od stranic tega objekta. Ob prenosu vseh 6 slik stranic objekta na strežnik, se na strežniškem delu sistema izvede še učenje detektorja in razvrščevalnika na novo dograjeni učni zbirki. Za učinkovitejše učenje smo dodali postopek samodejnega tvorjenja umetnih slik, ki temelji na izgradnji 3D modela objekta iz njegovih slik stranic in nam omogoča enostavno povečanje učnega materiala. Postopek izgradnje 3D modela iz slik stranic zahteva, da pri postopku registracije v grafičnem vmesniku ročno označimo vse štiri kote stranice na posamezni sliki, preostali koraki tvorjenja slik in učenja pa so samodejni. Na osnovi lokacije označenih kotov se s perspektivno transformacijo izvede poravnava stranice v frontalno lego. Model danega objekta nato sestavimo iz frontaliziranih slik vseh 6 stranic objekta. Umetne slike, ki jih uporabimo za učenje detektorja in razvrščevalnika, končno tvorimo z rotacijo modela v 3D prostoru in dodajanjem različnih ozadij, kot prikazuje slika 2.

2.4 Detekcija objektov

Za detekcijo objektov smo uporabili postopek FasterRCNN [1], implementiran v knjižnici CNTK [2]. Omenjeni postopek uporablja za detekcijo objektov konvolucijske nevronske mreže (implementacija trenutno podpira mreži VGG16 in AlexNet) in za učenje potrebuje veliko učnega materiala. Ker take količine učnih slik nimamo na razpolago smo vzeli globoko mrežo, ki je bila vnaprej naučena na zbirki slik ImageNet in nato le doučili parametre mreže na lastni zbirki slik. Postopek se je izkazal za hitrega in učinkovitega, z deležem detektiranih objektov preko 96% in povprečnim IOU (ang. intersection over union) enakim 0,78. Detekcijsko okno nam kasneje služi za kvadratni izrez področja, ki ga zahteva modul za izračun značilk.



(a) Slike stranic objekta z označenimi robnimi točkami.



(b) Frontalizirane slike stranic objekta.



(c) Umetno tvorjene slike

Slika 2: Postopek registracije in tvorjenja umetnih slik.

2.5 Izračun značilk

Ker postopki globokega učenja v zadnjih letih dosegajo visoke rezultate na področju računalniškega vida, smo se odločili, da izračun značilk izvedemo s pomočjo globokih mrež. V eksperimentalni primerjavi (poglavje 3.2) se je izkazalo, da največjo učinkovitost verifikacije zagotavljajo značilke, ki jih dobimo z mrežo Resnet kot izhode iz predzadnje plasti nevronov (izpustimo zadnjo polno povezano plast). Iz tega razloga smo se odločili, da za izračun značilk nadalje uporabljamo mrežo ResNet. Značilke izračunamo iz kvadratnega področja, ki zajema detekcijsko okno objekta, pri čemer smo preizkusili kakšen vpliv ima na končno uspešnost verifikacije ohranitev oz. odstranitev ozadja izven detekcijskega okna (poglavje 3.3).

2.6 Izračun podobnosti in odločanje

Dani objekt, ki ga posnamemo s kamero pametnih očal, primerjamo z izbranim registriranim objektom, ki smo ga predhodno izbrali v uporabniškem vmesniku. Na podlagi mere podobnosti med značilkami danega objekta in značilkami izbranega registriranega objekta se izvede sprejem oz. zavrnitev objekta, glede na to ali je izračunana podobnost večja ali manjša od vnaprej nastavljenega praga. Določitev praga in preizkus različnih vrst mer podobnosti eksperimentalno ocenimo v poglavju 3.4.

2.7 Uporabniški vmesnik

Komunikacija uporabnika s sistemom je omogočena preko več uporabniških vmesnikov. Uporabniški vmesnik na pametnih očalih je izveden v obliki spletne aplikacije, ki jo zaženemo v internetnem brskalniku očal in omogoča tako zajem slike kot prikaz rezultata verifikacije in prikaz detekcijskega okna na dani sliki objekta. Na ta način je zajem neodvisen od operacijskega sistema in razpoložljivosti razvojnega kompleta (SDK). Na skladiščnem računalniku dostopamo do več vmesnikov, od katerih eden omogoča registracijo objektov v sistem, drugi vmesnik nam omogoča izbiro objekta za katerega

želimo izvesti verifikacijo, tretji vmesnik pa predstavlja spletni arhiv slik z zavrženimi objekti. Vmesniki so implementirani v jeziku HTML5, CSS in JavaScript.

2.8 Komunikacijski kanali

Demonstracijski sistem deluje na treh enotah, ki vključujejo strežniški računalnik, kjer se izvaja glavna procesa verifikacije, pametna očala s pripadajočo spletno kamero ter skladiščni računalnik na katerem dostopamo do vmesnika za registracijo in izbiro objekta. Ko sistem deluje v načinu verifikacije je potrebno sprotno pošiljanje podatkov med pametnimi očali in skladiščnim računalnikom na eni strani ter strežnikom na drugi strani. V ta namen smo z uporabo spletnih vtičnic (ang. websockets) implementirali komunikacijska kanala, ki omogočata obojesmerni prenos podatkov. Komunikacijski kanal med vmesnikom pametnih očal in strežnikom skrbi za prenos slike v smeri do strežnika in za prenos rezultata verifikacije ter koordinat detekcijskega okna v obratni smeri. Na komunikacijskem kanalu med vmesnikom za izbiro objekta in strežnikom pa se v smeri proti strežniku vrši prenos oznake izbranega objekta, v obratni smeri pa teče prenos rezultata verifikacije.

2.9 Programska oprema

Jedro podpornega sistema, t.j. proces verifikacije, ki teče na strežniku, smo implementirali v programskem jeziku Python, pri čemer smo si pomagali z različnimi programskimi knjižnicami. S pomočjo knjižnice CNTK smo izvedli detekcijo objektov, knjižnici Tensorflow in Keras smo uporabili pri izračunu značilik, knjižnica Mayavi nam služi za 3D modeliranje objektov v procesu registracije, knjižnico Sklearn pa smo uporabili pri merjenju podobnosti med vektorji značilik. Pri udejanjenju komunikacijskih kanalov s strežnikom pa smo uporabili knjižnico Websockets. Spletni uporabniški vmesniki so implementirani s pomočjo programskih jezikov HTML5, CSS in JavaScript.

3 Eksperimenti

V tem poglavju predstavimo rezultate izvedenih eksperimentov. S pomočjo pridobljenih eksperimentalnih rezultatov smo nastavili oz. izbrali določene odprte komponente sistema, pri čemer smo težili k čim večji učinkovitosti verifikacije, ki smo jo merili v obliki deleža TPR pri 0,1% FPR. V okviru testiranja smo preverili tudi učinkovitost delovanja posameznih postopkov, ki smo jih integrirali v sistem, kot tudi delovanje prototipnega sistema kot celote. Eksperimente smo izvedli na lastni zbirki slik kosmičev, nekatere preliminarne eksperimente pa smo naredili na zbirki kosmičev, ki je rezultat diplomskega dela iz [3]. Ker sistem deluje v načinu verifikacije, smo uspešnost delovanja prikazali v obliki krivulj ROC (ang. receiver operating characteristic).

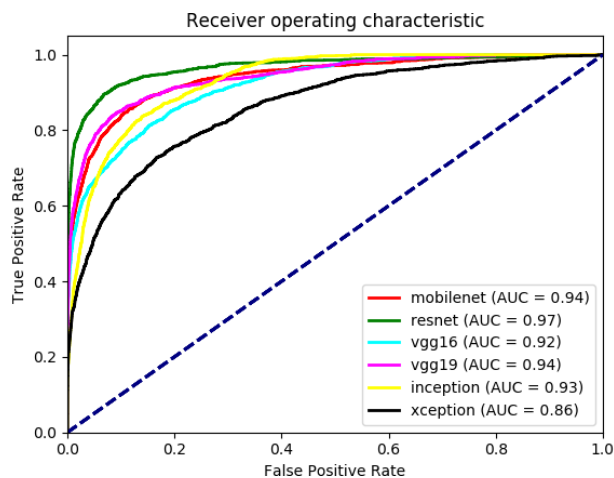
3.1 Snemanje lastne zbirke slik

Za potrebe evalvacije zanesljivosti razvitega sistema smo zbrali lastno zbirko slik 8 različnih škatel kosmičev, s katerimi smo poskusili imitirati okolje v katerem deluje

skladiščni delavec. Zbirka vsebuje testno množico in galerijsko množico. Testna množica vsebuje približno sto slik vsake od škatel, pri čemer smo ob njihovem zajemu težili k temu, da orientacija škatel na slikah čim bolj variira. Galerijsko množico poleg 6 slik stranic vsakega od 8 registriranih predmetov sestavlja 1000 umetnih slik vsakega od objektov, ki smo jih tvorili po postopku opisanem v poglavju 2.3.

3.2 Izbira globoke mreže za izračun značilik

V nizu eksperimentov, opisanem v tem razdelku, smo preizkusili uspešnost delovanja pri uporabi različnih prednaučenih globokih mrež, ki smo jih uporabili pri izračunu značilik. Globoke mreže potrebujejo velike količine podatkov za učinkovito učenje, ki je praviloma računsko zahtevno in posledično časovno potratno. Težavam pri učenju mrež smo se zato izognili z uporabo mrež, ki so bile predhodno naučene na velikem številu generičnih slik. Osredotočili smo se na primerjavo šestih prednaučenih mrež iz knjižnice Keras, ki so bile naučene na zbirki ImageNet [4]. Te mreže so VGG16 in VGG19 [5], Inception [6], Xception [7], ResNet [8] in MobileNet [9]. Rezultati teh eksperimentov so prikazani na sliki 3 iz katere lahko razberemo, da najvišjo uspešnost verifikacije dosežemo z mrežo ResNet. V vseh nadaljnjih eksperimentih smo zato za izračun značilik uporabili globoko mrežo ResNet.

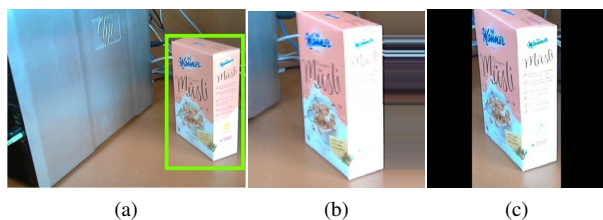


Slika 3: Uspešnost verifikacije ob uporabi različnih vrst prednaučenih globokih mrež.

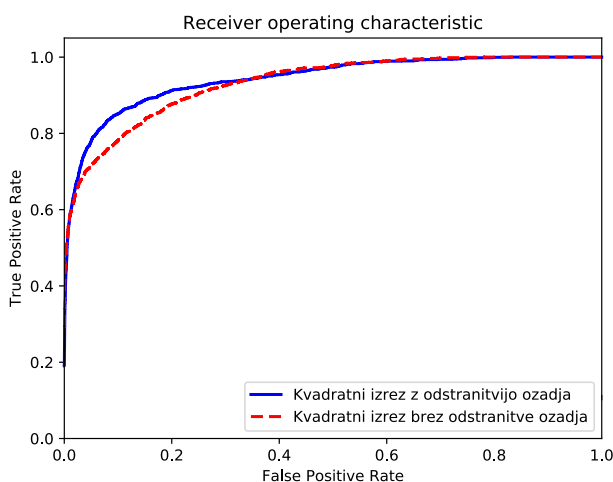
3.3 Definicija območja pri izračun značilik

Globoka mreža ResNet, ki jo uporabljamo pri izračunu značilik, zahteva na svojem vhodu sliko kvadratne oblike, dimenzije 229×229 . Detekcijski pravokotnik praviloma ni kvadratne oblike, ampak ima poljubno razmerje med višino in dolžino, ki je odvisno od oblike objekta na sliki. Obravnavali smo dve možnosti izvedbe kvadratnega izreza pravokotnega področja detekcije: *i*) kvadratni izrez, kjer ohranimo ozadje in ga po potrebi ekstrapoliramo, če je detekcijsko okno blizu roba slike (slika 4b), *ii*) kvadratni izrez, kjer vrednosti slikovnih elementov izven detekcijskega okna nastavimo na določeno konstantno vrednost (črno področje na sliki 4c). Izkazalo se je, da

slednji postopek prinese boljšo učinkovitost verifikacije (slika 5) v primerjavi s postopkom, kjer ohranimo ozadje.



Slika 4: Določitev kvadratnega območja pri izračunu značilk: a) Vhodna slika z detekcijskim oknom, b) kvadratni izrez brez odstranitve ozadja, c) kvadratni izrez z odstranitvijo ozadja.



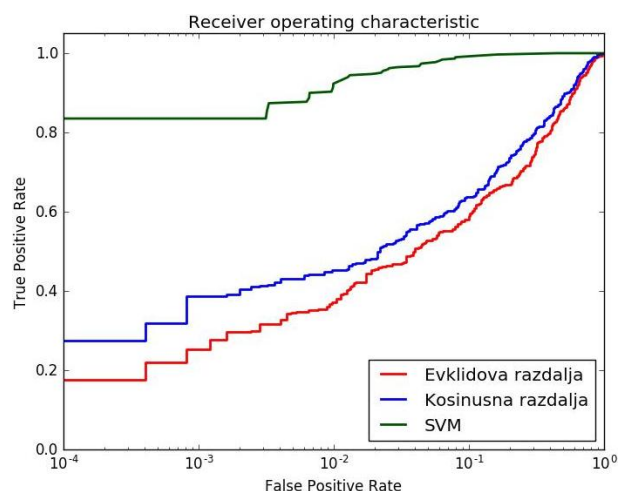
Slika 5: Uspešnost verifikacije glede na uporabljen postopek izreza.

3.4 Izbira mere podobnosti in razvrščevalnika

Pri izbiri mere podobnosti v modulu za odločanje smo, poleg iskanja najbližjega soseda iz vsakega razreda z uporabo Evklidove in kosinusne razdalje, preizkusili še merjenje podobnosti z metodo podpornih vektorjev [10] (ang. support vector machine, SVM), s katero smo, kot je razvidno iz slike 6, dosegli znatno izboljšanje uspešnosti verifikacije. Odločitveni prag smo določili glede na operacijsko točko, ki smo jo izbrali pri 0,1% napačnih sprejemov (ang. false positives). V tej točki je z uporabo metode podpornih vektorjev uspešnost verifikacije preseгла 80% pravih sprejemov (ang. true positives).

4 Zaključek

V članku smo predstavili demonstracijski sistem za skladiščne delavce, ki smo ga razvili v okviru projekta GOSTOP. Podporni sistem omogoča pohitritev opravi skladiščnih delavcev in enostavnejše preverjanje istovetnosti produktov v skladiščih. Opisali smo osnovne značilnosti razvite storitve, opisali njeno arhitekturo in funkcionalnost ter eksperimentalno ovrednotili njeno delovanje. Pomembna lastnost omenjenega sistema je, da ga je možno uporabiti z vsako napravo, ki podpira novejšje spletne brskalnike (v našem primeru Firefox).



Slika 6: Uspešnost verifikacije pri različnih postopkih merjenja podobnosti.

Zahvala

Raziskovalno delo, ki je pripeljalo do predstavljenih rezultatov, je bilo delno financirano s strani ARRS (raziskovalna programa Metrologija in biometrični sistemi, P2-0250, in Računalniški vid, P2-0214) ter programa GOSTOP (št. pogodbe C3330-16-529000), financiranega s strani RS-MIZŠ in EU-ESRR.

Literatura

- [1] R. Girshick, "Fast R-CNN," v *Proc. of the 2015 IEEE Int. Conf. on Comp. Vis. (ICCV)*, ser. ICCV '15. Washington, DC, ZDA: IEEE Computer Society, 2015, str. 1440–1448.
- [2] F. Seide in A. Agarwal, "CNTK: Microsoft's Open-Source Deep-Learning Toolkit," v *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, ZDA: ACM, 2016, str. 2135–2135.
- [3] U. Brodnik, "Detekcija artiklov na prodajnih policah z računalniškim vidom," *Univerza v Ljubljani, Fakulteta za elektrotehniko*, zv. Diplomsko delo, 2012.
- [4] J. Deng *et al.*, "ImageNet: A large-scale hierarchical image database," v *2009 IEEE Conf. on Comp. Vis. and Pat. Rec.*, Jun. 2009, str. 248–255.
- [5] K. Simonyan in A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR*, 2014.
- [6] C. Szegedy *et al.*, "Going Deeper with Convolutions," v *2015 IEEE Conf. on Comp. Vis. and Pat. Rec. (CVPR)*, Jun. 2015, str. 1–9.
- [7] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," v *2017 IEEE Conf. on Comp. Vis. and Pat. Rec. (CVPR)*, Jul. 2017, str. 1800–1807.
- [8] K. He, X. Zhang, S. Ren, in J. Sun, "Deep Residual Learning for Image Recognition," v *2016 IEEE Conf. on Comp. Vis. and Pat. Rec. (CVPR)*, June 2016, str. 770–778.
- [9] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *CoRR*, 2017.
- [10] C. Cortes in V. Vapnik, "Support-Vector Networks," *Mach. Learn.*, zv. 20, št. 3, str. 273–297, Sept. 1995.