Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Barbara Aljaž

Učenje konvolucijskih nevronskih mrež iz sintetičnih podatkov na primeru detekcije rok

diplomsko delo

univerzitetni študijski program
prve stopnje
računalništvo in informatika

Mentor: doc. dr. Luka Čehovin Zajc

Ljubljana, 2018

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V okviru diplomske naloge preučite možnost učenja konvolucijske nevronske mreže za detekcijo rok v video tokovih z uporabo sintetičnih učnih podatkov. Za detekcijo izberite ustrezno arhitekturo nevronske mreže ter implementirajte avtomatsko generiranje in anotiranje sintetičnih učnih podatkov. Naučeni detektor ovrednotite na ustreznih zbirkah realnih slik.

Za pomoč in podporo pri izdelavi te diplomske naloge se zahvaljujem mentorju, osebju laboratorija ViCoS ter prijateljem in družini.

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
3D	three-dimensional	tridimenzionalni
CNN	convolutional neural network	konvolucijska nevronska mreža
HOG	histogram of gradients	histogram usmerjenih gradientov
IOU	intersection over union	razmerje med presekom in unijo
LDA	linear discriminant analysis	linearna diskriminantna analiza

Povzetek

Naslov: Učenje konvolucijskih nevronskih mrež iz sintetičnih podatkov na primeru detekcije rok

Avtor: Barbara Aljaž

Za učenje konvolucijskih nevronskih mrež je potrebna velika količina podatkov, ki jih je potrebno pridobiti in anotirati. Pogosto se za povečanje učnih zbirk uporabljajo različne augmentacije, mi pa smo v tem diplomskem delu raziskali možnost uporabe umetno generiranih podatkov. Ustvarili smo jih na podlagi tridimenzionalnega modela in parametre, ki so vplivali na zajete slike, nadzorovali avtomatsko. Delovali smo na primeru zaznave človeških rok in detektor preizkusili na dveh zbirkah realnih slik v okviru scenarija brezdotične interakcije med človekom in računalnikom. Primerjali smo ga z detektorjem, naučenim iz realističnih podatkov in analizirali razlike. Rezultati predstavljenega eksperimenta so obetavni in nakazujejo več možnosti za nadaljnji razvoj take vrste učenja.

Ključne besede: računalniški vid, konvolucijske nevronske mreže, YOLO, umetno generirani podatki.

Abstract

Title: Using Synthetic Data to Train Convolutional Neural Networks for the Case of Hand Detection

Author: Barbara Aljaž

Convolutional neural networks require a large amount of data for training that need to be collected and annotated. Methods used to enlarge learning dataset usually include different augmentations, but in this thesis we researched the possibility of using artificially generated data samples. We created them using a three dimensional model and automatically controlled parameters that influenced captured images. We worked on the example of human hand detection and evaluated our detector on two datasets of real images for a touch-less interface human-computer interaction scenario. We compared it with a detector trained on real life data and analyzed the differences. Results of the experiment are promising and present many opportunities for further development of such training technique.

Keywords: computer vision, convolutional neural networks, YOLO, synthetic data.

Poglavje 1

Uvod

V zadnjih letih veliko rešitev problemov na področju računalniškega vida, kot so detekcija, klasifikacija ali segmentacija, temelji na globokem učenju. To je posledica več faktorjev, med drugim večje računske moči sodobnih računalnikov in velike količine podatkov, ki so nam na voljo. Učenje globokih nevronskih mrež pogosto zahteva podatkovne zbirke, ki vsebujejo nad 1000 primerkov. Podatke, v našem primeru slike, je potrebno pridobiti (zajeti ali poiskati v obstoječih virih), temu pa sledi dolgotrajna anotacija. Možnosti je več, podatke lahko anotira znanstvenik sam, najame nekoga, ki to naredi namesto njega, ali pa, kot so se tega problema lotili na družabnem omrežju Facebook, anotacijo naredijo kar uporabniki sami (Facebook spodbuja uporabnike, da sami označijo, kdo se nahaja na sliki). V vsakem primeru je to del učnega postopka, za katerega je potrebno veliko ur človeškega dela. Zato si znanstveniki pomagajo z umetnim povečevanjem podatkovnih zbirk. Najbolj priljubljena je uporaba različnih augmentacij. Na primeru slik to pomeni, da se originalne slike obdelajo z afinimi transformacijami, dodajanjem šuma, meglenjem ipd. Tako se količina učnih podatkov poveča.

V zadnjem času se vse pogosteje uporablja umetno generiranje podatkov, saj sodobna grafika omogoča oblikovanje zelo realističnih tridimenzionalnih modelov. Z njimi je možno računalnik naučiti reševati probleme iz različnih

domen. Mi smo se usmerili na problem detekcije dlani. Njeno premikanje in videz lahko s tridimenzionalnim modelom kar precej točno modeliramo. Poleg tega je to aktualna tema, uspešni lokalizaciji roke v naslednjem koraku namreč sledi prepoznavanje gest. To je problem, s katerim se znanstveniki ukvarjajo že zelo dolgo. Prepoznavanje gest rok nam med drugim omogoča brezstično komunikacijo med človekom in računalnikom. To odpira možnosti sporazumevanja v primerih, ko si želimo hitre in čim bolj preproste komunikacije. S tem diplomskim delom smo zajeli dve trenutno aktualni področji raziskovanja in ju povezali v delujoč sistem, ki odpira vrata v nadaljnje raziskovanje.

Diplomo sestavlja pet poglavij: uvod, pregled področja, metodologija, eksperimentalno ovrednotenje in zaključek. V drugem poglavju smo se na kratko dotaknili zgodovine in trenutnih smernic razvoja detekcije rok in uporabe umetno generiranih podatkov. V tretjem poglavju smo opisali nevronske mreže kot način učenja, uporabljeno nevronsko mrežo in postopek zajemanja podatkov ter priprav na učenje. V četrtem poglavju smo predstavili naučene modele in rezultate. V zadnjem poglavju smo povzeli prispevke tega diplomskega dela.

Poglavje 2

Pregled področja

Detekcija dlani, ki ji v naslednjem koraku sledi prepoznavanje gest, je aktualno področje zaradi veliko možnosti uporabe sistemov, ki uporabljajo brezstično komunikacijo. Primer takšnega sistema so vozila, kjer hitra komunikacija med človekom in računalnikom omogoča vožnjo z malo motnjami. Uporaba brezstične komunikacije se ne konča pri vozilih, ampak jo lahko, ker so geste z rokami pomemben del naravne človeške komunikacije, povežemo z mnogimi domenami (na primer oddaljen nadzor sistemov, obogatena resničnost, video igre in znakovni jezik).

Prav tako je umetno generiranje podatkov v zadnjem času precej aktualna tema in zaželen način pridobivanja podatkov na področju računalniškega vida, saj zmanjša količino potrebnega človeškega dela in ponudi večji nadzor nad parametri, ki vplivajo na učne podatke (na primer osvetlitev).

2.1 Lokalizacija dlani in detekcija gest

V tem poglavju smo na kratko povzeli pregled zgodovine in trenutnih trendov na področju detekcije dlani in razpoznavanja gest rok.

2.1.1 Načini lokalizacije roke

Začetni korak prepoznavanja gest je lokalizacija roke. Znanstveniki so se tega problema lotili na različne načine.

Barva

Segmentacija kožne barve je najbolj očiten in pogosto uporabljen ([11, 27, 39]) način lokalizacije roke, ker lahko ob predpostavki, da se na sliki nahaja tudi obraz, uporabimo kar njegovo barvo [3]. Prvi korak segmentacije predstavlja izbira barvnega prostora. Zaželeno je, da sta v njem barvna in svetlobna komponenta ločeni (primeri takšnih barvnih prostorov so HSV, YCrCb in YUV). Tako dosežemo, da je detekcija barve manj odvisna od osvetljenosti in senc. Po izločitvi komponente osvetlitve dobimo dvodimenzionalni vektor, ki je skoraj identičen za različne slike kože. Ko iz več slik izločimo barve histograme in jih statistično obdelamo, dobimo model kožne barve, ki ga lahko uporabimo za segmentacijo na drugih slikah. Slabosti takšnega načina lokalizacije dlani sta občutljivost na hitre spremembe v svetlobi in možne napačne zaznave, če so na sliki tudi drugi objekti podobne barve kot je dlan. Zato se barva kot način lokalizacije dlani ponavadi uporablja v kombinaciji z drugimi metodami, kot so to storili v [22].

Oblika

Zaradi značilne oblike dlani lahko že z enostavno izločitvijo konture iz slike roke na enobarvnemu ozadju dobimo dober približek, kje se ta nahaja. Ker pa v splošnem ozadje ni enobarvno in lahko pride do problema prekrivanja, pri zaznavi, ki temelji na robovih, dobimo tudi robove, ki hkrati pripadajo roki ter ozadju. Zato je nujno, da izvedemo tudi dodatno obdelavo pridobljene konture, ali pa da ozadje vnaprej izločimo. Bolj smiselno je opisati obliko roke z HOG deskriptorji (to so storili v [1, 6, 9]). Te dobimo tako, da sliko odvajamo, jo razdelimo na manjše pravokotnike in nato v vsakem izmed teh polj izračunamo histogram gradientom ter ga normaliziramo. Kot deskrip-

torji so ti histogrami invariantni na manjše premike, rotacije in raznolikost v kontrastu.

Tekstura

V [2, 14, 19] so se usmerili v to, da bi segmentacijo roke izvedli glede na njen videz in teksturo. Razlog je tudi v tem, da barvne slike niso vedno na voljo. Pri tem so predpostavili, da se roka v različnih izvajanih gestah razlikuje bolj kot se razlikujejo roke različnih oseb. Največji problem pri tem načinu je avtomatska izločitev značilk, kar je prispevalo k temu, da detekcija, ki temelji na vrednostih slikovnih elementov, do začetka enaindvajsetega stoletja ni dosegala vidnejših rezultatov. V zadnjih letih so znanstveniki našli več rešitev opisanega problema. Med njimi so tudi konvolucijske nevronske mreže, ki se o nahajanju roke v neki regiji odločajo glede na konvolucijo z različnimi filtri.

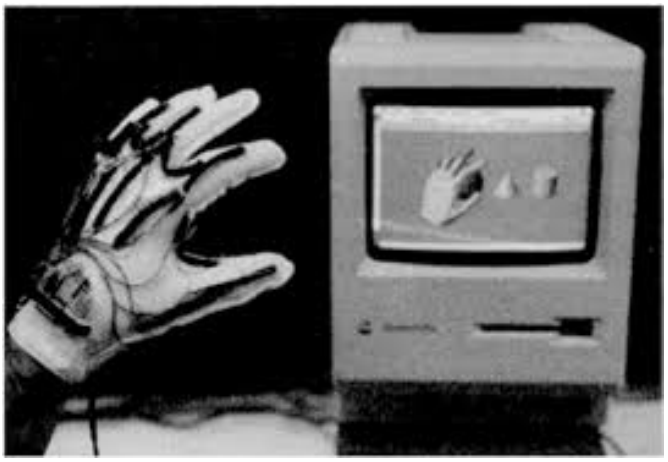
Globina

Izkazalo se je, da najboljše rezultate učenja razpoznavanja gest dosežemo z uporabo globinske kamere v kombinaciji z RGB kamero (na primer sistem Kinect), kot so to storili v [12, 17, 24]. Pogosto pa lokalizacija regije roke temelji na informaciji o barvi, ki je občutljiva na osvetlitev ali pa na predpostavki, da je roka najbližji objekt v globinski sliki, kar je za primere direktne komunikacije med človekom in računalnikom sicer večinoma ustrezno, v splošnem primeru detekcije rok na sliki pa ni nujno res. Poleg tega je slabost sistemov, ki temeljijo na globinskih senzorjih tudi v uporabi infrardeče svetlobe. Njena zaznava je namreč zunaj nadzorovanih prostorov lahko motena, na primer zaradi sončne svetlobe.

2.1.2 Zgodovina raziskav detekcije gest rok

Začetek poskusov brezstične komunikacije človek - računalnik seže v sedemdeseta leta prejšnjega stoletja, ko so znanstveniki skušali razviti rokavice s senzorji, ki bi omogočali zajem premikov uporabnikove roke. Eden izmed

zgodnjih prototipov, Sayre Glove (razvit leta 1977), je bil sestavljen iz elastične cevi, ki je imela na eni strani vir svetlobe, na drugi strani pa fotocelice. Deloval je tako, da je zaznaval krčenje prstov glede na količino zajete svetlobe. V naslednjih letih so se dodajali senzorji, ki so še izboljšali zaznavo premikov roke, vendar pa so bile aplikacije takšnih rokavic precej ozko usmerjene in so služile zgolj za raziskovalne namene. Primer rokavice, ki deluje na podlagi senzorjev se nahaja na Sliki 1.



Slika 1: The Z^{TM}Glove, ki jo je leta 1982 razvil Zimmerman [26, Slika 2.2]

Prva komercialno dostopna rokavica je prišla na trg leta 1983 in sprožila razvoj različic, ki so se uporabljale za namene igranja računalniških iger in so jih sestavljali različni senzorji. Razvoj rokavic za zaznavo premikov roke se nadaljuje še danes in obravnava različna področja, med drugim tudi zdravstvo.

V zgodnjih letih razvoja prepoznavanja gest so imele obstoječe kamere nizko resolucijo, računalniki pa manjšo procesno moč kot danes, zato zaznava dlani in gest z uporabo računalniškega vida ni veljala za zanesljivo. Kljub temu so bili prvi sistemi, ki so temeljili na računalniškem vidu, razviti v začetku osemdesetih let prejšnjega stoletja. Eden izmed vidnejših napredkov takšne zaznave so bile rokavice, ki so delovale pasivno - na njih so bile barvne oznake,

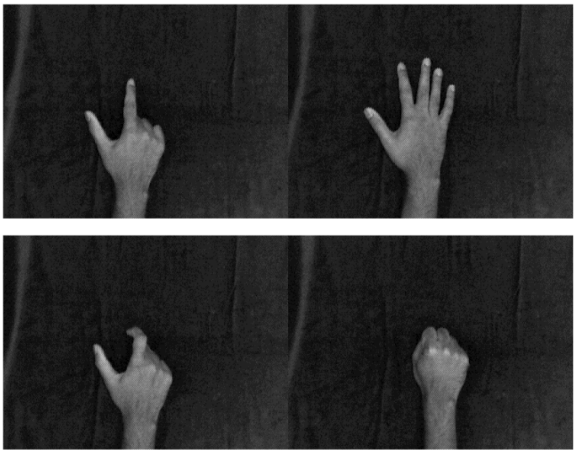
ki jih je zaznala zunanja kamera in tako predvidevala potek gibanja roke. Natančnost sistemov, v ozadju katerih je bilo delovanje na ta način, ni bila visoka, sploh zaradi možnega prekrivanja oznak in različnega izvajanja gest različnih uporabnikov.

Z izboljšanjem kamer in računalnikov se je razvoj sistemov detekcije gest preusmeril k uporabi računalniškega vida, ki ne zahteva uporabe rokavic, ki bi omejevale uporabnika in je cenovno ugodnejši. Prvi sistem, ki je temeljil izključno na zaznavi gibanja dlani brez oznak, je bil razvit leta 1993 in je bil sposoben sledenja s frekvenco 10 Hz (DigitEyes [30]). Za zaznavo gibanja roke se je uporabljal model, ki je slonel na točkovnih in črtnih značilkah, izločenih iz črno-belih slik rok. Posebnosti tega sistema sta bili tudi delovanje v sedemindvajsetih stopnjah svobode in uporaba dveh kamer za preprečevanje napačne zaznave v primeru prekrivanja. Robustnost detekcije roke je leta 1995 še izboljšal Freeman z uporabo HOG deskriptorjev [6].

V devetdesetih letih se je natančnost sistemov za sledenje roke in detekcije gest še izboljševala. Večje zanimanje je bilo namenjeno tudi detekciji dinamičnih gest, kar je do takrat veljalo za težek problem. Poleg tega sta leta 1998 Segen in Kumar [34] razvila sistem, ki je omogočal prepoznavo štirih gest (prikazane na Sliki 2), sledil roki v tridimenzionalnem prostoru in deloval s 60 sličicami na sekundo. Uporabljal se je za splošno komunikacijo človek - računalnik in igranje iger ter povzročil razvoj mnogih podobnih vmesnikov za brezstično komunikacijo.

2.1.3 Trenutni trendi

V začetku dvajsetega stoletja je bilo razvitih kar nekaj algoritmov, ki so obravnavali lokalizacijo in prepoznavo gest rok. Klasifikacija v več fazah je bila leta 2004 predlagana v [14]. Detektor je temeljil na metodi AdaBoost in je bil sestavljen iz velikega števila šibkih klasifikatorjev, odločanje ali okno vsebuje roko in če, katero gesto, pa je potekalo v drevesni strukturi.



Slika 2: Geste, ki jih je prepoznaval sistem, ki sta ga razvila Segen in Kumar [34, Slika 4]

Druga, na področju računalniškega vida pogosto uporabljena metoda, ki so jo prilagodili za detekcijo gest rok, je Random forest v kombinaciji z linearno diskriminantno analizo [25]. Naključno zgrajena odločitvena drevesa slonijo na lokalnih razlikah v intenziteti in izločijo potencialne regije, nad katerimi se za odločitev, ali regija vsebuje roko in če, katero gesto, izvede algoritem LDA.

Ob širitvi uporabe konvolucijskih nevronskih mrež na raznolike domene računalniškega vida se je kot potencialna izkazala tudi uporaba le-teh za detekcijo in sledenje rok [23, 24, 38, 40].

2.2 Umetno generirani podatki

V zadnjih letih se na področju računalniškega vida kot alternativa augmentaciji za večanje učnih množic uporablja tudi umetno generiranje podatkov. Glavna prednost, ki smo jo tudi mi izkoristili, je manjša količina ur potrebnega dela, predvsem za anotacijo. Prvi primer umetnega generiranja podatkov [33] je bil posledica želje po zaščiti osebnih podatkov in se je uporabil na primeru štetja prebivalstva. V naslednjih letih se je izkazalo, da imajo

sintetični podatki tudi druge prednosti. Njihova uporaba se je razširila na različne domene. V naslednjih podpoglavjih bomo obravnavali zgodovino in sodobne trende učenja iz sintetičnih podatkov.

2.2.1 Področja uporabe sintetičnih podatkov

Na našem primeru smo pokazali, da lahko umetno generirane podatke uspešno in s primerljivi rezultati glede na realistične podatke uporabimo za učenje konvolucijske nevronske mreže. V praksi je pogosto težko generirati sintetične podatke (na primer medicinske slike), zato se umetno generirani podatki v večini primerov uporabljajo v kombinaciji z realističnimi. Domeno detekcije roke smo izbrali, ker je dlan in njeno gibanje lahko precej enostavno poustvariti s tridimenzionalnim modelom, ker se človeške roke ne razlikujejo preveč. Domena je na mnogih področjih preširoka, recimo če bi želeli učiti detektor mačk iz sintetičnih podatkov, bi bilo to zaradi raznolikosti barv in vzorcev mačje dlake precej zahtevna naloga. Kljub temu lahko dosežemo zadovoljive rezultate učenja detekcije iz tridimenzionalnih modelov, tudi v primeru raznolikih razredov. To so pokazali v [35] na 20 različnih kategorijah, ki so prikazane na Sliki 3.

Slika 3: Primeri tridimenzionalnih modelov uporabljenih za učenje detekcije [35, Slika 2]

To je le eden od uspešnih primerov uporabe umetno generiranih podatkov na področju računalniškega vida. Sintetični podatki so bili uporabljeni tudi na drugih domenah, kot so rekonstrukcija obraza iz slike [31] ali segmentacija na podlagi podatkov iz računalniških iger [32].

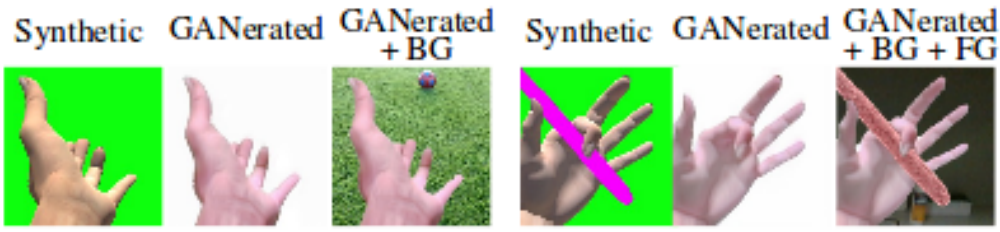
2.2.2 Uporaba sintetičnih podatkov na področju zaznave dlani

Kot je bilo omenjeno v prejšnjem poglavju, je s tridimenzionalnimi modeli možno ustvariti slike raznolikih predmetov. Tako lahko avtomatizirano ustvarimo učno množico slik, ne da bi bilo potrebno slike anotirati ročno. Znan problem na področju človeškega anotiranja tridimenzionalnih podatkov je med drugim nenatančnost. V [40] so se zaradi tega in takratnega pomanjkanja podatkovnih baz, ki bi vsebovale označbe poze na barvnih slikah, zajetih z monokularno kamero, učenja ocenjevanja poze, v kateri je roka, lotili s sintetičnimi podatki. Primer uporabljene umetno generirane slike se nahaja na Sliki 4.

Slika 4: Primer iz podatkovna množice, ki ga sestavljata segmentacijska maska z 33 razredi (trije za vsak prst, dlan, oseba in ozadje) ter tridimenzionalni model, ki vsebuje 21 značilk za vsako roko (štiri značilke za vsak prst in eno značilko blizu zapestja) [40, Slika 4]

Ocenjevanje poze, v kateri se nahaja roka, je potekalo v treh fazah. Najprej so dlan lokalizirali s segmentacijsko nevronsko mrežo, z masko iz celotne slike izrezali sliko dlani in jo posredovali naslednji nevronski mreži, katere naloga je lokalizacija značilk. Te značilke so s tretjo nevronsko mrežo primerjali z naučenimi značilkami in glede na podobnost ocenili v kateri pozi je dlan. Ta raziskava je dodala še dodaten korak (ocenjevanje poze) po lokalizaciji roke. Od drugih podobnih raziskav se razlikuje v tem, da ni bila uporabljena globinska kamera, ampak so parametre ocenjevali iz dvodimenzionalnih slik.

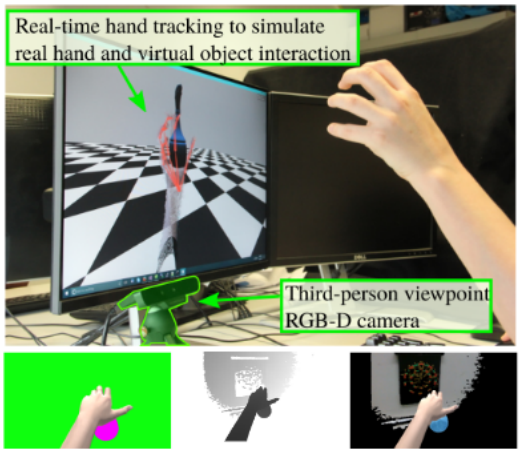
Pomanjkljivost [40] je v tem, da je detekcija zelo občutljiva na prekrivanja, poleg tega pa se pri učenju iz izključno sintetičnih podatkov lahko zgodi, da ti slabo generalizirajo realni svet. To so v [23] reševali z izvedbo translacije med sintetičnimi in realističnimi podatki ter na podlagi tega generirali učne slike (primeri tako generiranih slik se nahajajo na Sliki 5). Nato so učili nevronsko mrežo, imenovano RegNet (temelji na modelu ResNet [10]). Njena naloga je bila izločitev dvodimenzionalnih in tridimenzionalnih pozicij 21-ih sklepov roke. Na osnovi teh pozicij so ocenili pozo v kateri je roka. Rezultat raziskave je detekcija poze roke v realnem času, ki pa še vedno deluje slabše kot pri metodah, ki uporabljajo globinsko kamero.



Slika 5: Primeri umetno generiranih slik, ki so jih izboljšali z realnimi slikami rok ter jim dodali ospredje/ozadje [23, Slika 5]

Precejšnje izboljšanje na področju detekcije roke in prepoznavanju gest je ponudila uporaba globinskih senzorjev kot na primer v sistemu Kinect. V zadnjih letih zato veliko metod temelji ravno na uporabi le-teh. V [24] so v

nasprotju z veliko drugimi raziskavami konvolucijsko nevronsko mrežo učili s prvoosebnimi umetno generirami slikami. Postopek ocenjevanja poze roke so izvedli z dvema mrežama. Naloga prve mreže je bila poiskati lokacijo rok, naloga druge pa regresija pozicij sklepov rok. Umetno generirano učno množico so pridobili z zajemom gibanja roke različnih uporabnikov, ki so ga nato preslikali na fotorealističen tridimenzionalni model roke (zajem gibanja je prikazan na Sliki 6). Tako pridobljena podatkovna množica je bolj realistično predstavljala gibanje človeške roke, kar pa so še izboljšali z dodajanjem različnih odtenkov barve kože in anatomičnih značilnosti, kot so na primer dlake. Modelu roke so dodali tudi razne virtualne predmete, da bi poustvarili pojave, kot je to, da roka drži nek predmet, in realistična ozadja.

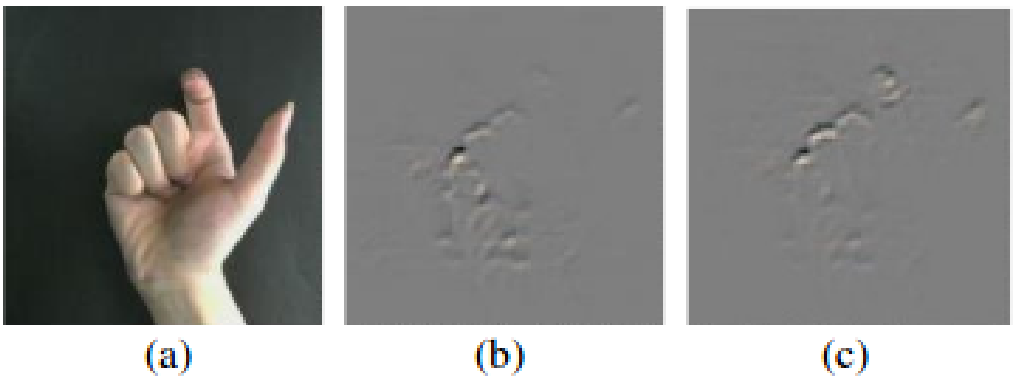


Slika 6: Zajem gibanja in preslikava na tridimenzionalni model roke. [24, Slika 5]

V raziskavi [38] so preučevali problem prepoznavanja 24-ih različnih gest rok. Oblikovali so tridimenzionalni model roke in ga premikali ter zajemali slike z orodjem Blender[1]. Uporabili so konvolucijsko nevronsko mrežo, ki je temeljila na mreži AlexNet [16]. Pokazali so, da je mogoče s kombinacijo velike množice sintetičnih in majhnega deleža realističnih slik doseči boljše rezul-

[1]https://www.blender.org/

tate kot pri uporabi izključno umetno generiranih podatkov. Raziskali so tudi vpliv uporabe sintetičnih in realističnih slik na konvolucijske in polnopovezane nivoje mreže. Eksperimentalno so ugotovili, da dodajanje realističnih slik v učno množico veliko bolj vpliva na uteži polnopovezanih plasti, medtem ko so uteži konvolucijskih plasti pri modelu, ki uporablja kombinacijo realističnih in sintetičnih slik, zelo podobne utežem modela, ki se uči iz izključno sintetičnih podatkov. Na Sliki 7 je prikazan primer geste, ki jo model, ki se uči iz izključno umetno generiranih podatkov, ne prepozna, pravilno pa jo prepozna model, ki se uči iz kombiniranih podatkov. Če primerjamo prikaz uteži prve polnopovezane plasti obeh modelov ugotovimo, da ima model, ki se uči iz sintetičnih in realističnih podatkov, (c) višje uteži na linijah prstov, predvsem kazalca, kar prispeva k boljši razpoznavi.



Slika 7: Primerjava uteži polnopovezanih plasti [24, Slika 5].

Da je umetno generiranje slik rok aktualno in ima tržni potencial kaže dejstvo, da obstaja spletna stran, na kateri lahko kupiš sintetično učno množico slik rok[2].

[2]https://www.datagen.tech

Poglavje 3

Metodologija

V zadnjih letih se je povečala uporaba konvolucijskih nevronskih mrež za reševanje problemov na področju računalniškega vida. Zaradi njihove aktualnosti smo se odločili, da preizkusimo njihovo uspešnost na področju detekcije roke. Pri tem smo se usmerili na učenje iz sintetičnih podatkov. V tem poglavju sta obravnavani dve temi, ki sta glavni komponenti naše raziskave, nevronske mreže, opisane v splošnem in usmerjeno v naš problem ter umetno generiranje učnih slik.

3.1 Konvolucijske nevronske mreže

Konvolucijske nevronske mreže (CNN) so vrsta umetnih nevronskih mrež, ki je specializirana za obdelavo podatkov, ki so organizirani v polja (na primer slike). Njihov razvoj je precej zaznamovala nevroznanost, predvsem delovanje primarnega vidnega korteksa (V1), dela možganov, ki je posvečen zgodnji napredni obdelavi vizualnih dražljajev. Kljub podobnostim med našimi možgani ter konvolucijskimi nevronskimi mrežami so te le približek delovanja možganov.

Konvolucijske nevronske mreže so se začele razvijati v 80-ih letih prejšnjega stoletja. Kot prva konvolucijska nevronska mreža se obravnava neokogni-

tron, ki ga je leta 1980 postavil Fukushima [7]. Arhitektura sodobnih nevronskih mrež temelji na neokognitronu, vendar pa, ta za razliko od njih, ni uporabljal metode vzvratnega učenja, ampak mednivojsko nenadzorovano gručenje. Metoda vzvratnega učenja je bila uspešno uporabljena leta 1989 v nevronski mreži, ki je prepoznavala ročno napisane številke in delovala na podlagi konvolucije [18]. Ta raziskava, ki jo je s sodelavci izvedel LeCun, predstavlja začetek sodobnega razvoja konvolucijskih nevronskih mrež.
Razvoj konvolucijskih nevronskih mrež se je nadaljeval, vendar pa do leta 2012, ko je Alex Krizhevsky z mrežo Alexnet [16] zmagal na ImageNet tekmovanju, niso dosegale tako dobrih rezultatov kot druge nevronske mreže.
V naslednjih letih se tipični gradniki konvolucijskih nevronskih mrež niso spreminjali, zaradi bolj zmogljive strojne opreme pa so se razvijale nove arhitekture. Ker z večjim številom parametrov lahko shranimo več informacij, globlje konvolucijske nevronske mreže pogosto dosegajo boljše rezultate.

3.1.1 Gradniki konvolucijskih nevronskih mrež

Konvolucijske nevronske mreže so organizirane v plasti, ki opravljajo različne funkcije, opisane v nadaljevanju.

Konvolucijska plast

Konvolucijske nevronske mreže temeljijo na operaciji konvolucija. V splošnem je konvolucija operacija dveh funkcij, ki kot rezultat proizvede tretjo funkcijo. Zapišemo jo kot:

$$s(t) = (x * w)(t) = \int x(a)w(t-a)da$$

pri čemer x označuje vhodno funkcijo, w pa utežitveno funkcijo (v terminologiji nevronskih mrež bi ju imenovali vhod in jedro). Ko podatke obdelujemo z računalniki, so ti v večini primerov diskretni, zato lahko konvolucijo definiramo tudi kot:

$$\sum_{a=-\infty}^{\infty} s(t) = x(a)w(t-a)$$

V konvolucijskih nevronskih mrežah so vhod konvolucije večdimenzionalne matrike podatkov, jedro pa je prav tako večdimenzionalno in vsebuje parametre, ki jih uporablja učni algoritem. Ker so te matrike shranjene v spominu, ki je končne velikosti, lahko predpostavimo, da so različne od nič le na nekem intervalu. Poleg tega konvolucijo, v primeru večdimenzionalnih podatkov, večinoma izvajamo na več oseh hkrati. Na primer, če je vhod dvodimenzionalna slika I in je jedro K prav tako dvodimenzionalno, konvolucijo zapišemo kot:

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n)$$

Ker pa je konvolucija komutativna, lahko enakovredno zapišemo:

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n)$$

in tako dobimo manj spremenljivosti v množici možnih vrednosti m in n.

Z uporabo konvolucije dosežemo manjšo prostorsko zahtevnost kot pri tradicionalnih nevronskih mrežah in učenje, ki je neodvisno od lokacije v sliki (isti filtri se uporabljajo za celo sliko). Poleg tega omogoča obdelavo vhodnih podatkov različnih velikosti.

Aktivacijska funkcija

Ker želimo, da je izhod iz konvolucijske plasti v nekem razponu (na primer med nič in ena), pogosto po izvedeni konvoluciji rezultat pretvorimo z eno izmed aktivacijskih funkcij. V konvolucijskih nevronskih mrežah je najpogosteje uporabljena aktivacijska funkcija ReLU (ang. *Rectified Linear Unit*), ki vhod upraguje z 0. Zapišemo jo lahko kot:

$$f(x) = max(0, x)$$

Prednost ReLU funkcije je pospešenem pojavu konvergence stohastičnega gradientnega sestopa in to, da je njena implementacija v primerjavi z *tanh* ali sigmoidno funkcijo enostavna in ne vsebuje računsko zahtevnih operacij. Uporaba ReLU funkcije pa ima tudi slabost, pojav ko zaradi visokega gradienta nevron „umre“in se ne bo nikoli več aktiviral. To poskuša reševati prepustni ReLU (ang. *leaky ReLU*), ki namesto nastavitve negativnih vrednosti na nič, doda funkciji majhen naklon.

Združevalna plast

Združevalna plast nadomesti izhod iz aktivacijske funkcije na neki lokaciji z rezultatom neke statistične operacije med trenutno in sosednjimi lokacijami. V našem primeru smo uporabili združevanje glede na maksimalno vrednost (ang. *max pooling*). To pomeni, da se je izhod nadomestil z največjo vrednostjo v okolici. Namen združevalne plasti je doseči invariantnost na majhne premike. Poleg tega lahko izvedemo združevanje z uporabo razmika in posledično dobimo kot rezultat polje, ki je manjše kot izhodno polje aktivacijske funkcije. Tako pridobimo na učinkovitosti računanja ter lahko s prilagajanjem razmika obdelujemo slike različnih velikosti.

Ostale uporabljene plasti

Te plasti se uporabljajo zgolj med učenjem z namenom reševanja določenih problemov, med inferenco pa se jih spusti.

Normalizacija skupin (ang. *batch normalization*) rešuje problem pojava kovariančnih premikov (sprememba porazdelitve vhodov v plast ob modifikaciji prejšnje plasti). Z globino je ta pojav vedno bolj opazen in plasti se morajo zato nenehno prilagajati spremembi porazdelitve, kar upočasnjuje učenje. Normalizacija skupin omogoča višjo učno stopnjo in regularizira učenje.

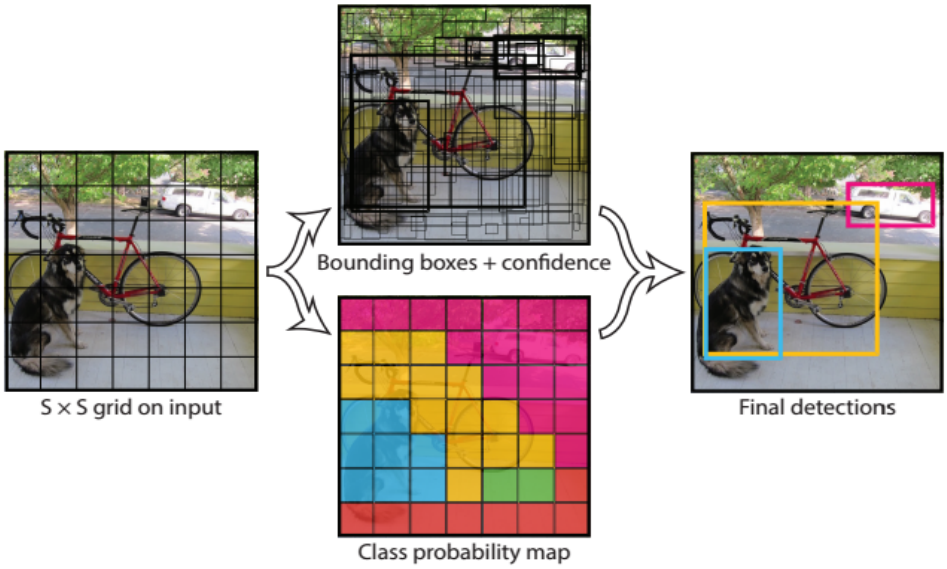
Izpuščanje nevronov (ang. *dropout*) je plast, ki se uporablja za preprečevanje pretiranega prilagajanja učnim podatkom (ang. *overfitting*). Naključno izbere ne-vhodne nevrone in jih izključi (njihov izhod se pomnoži z nič). Izključevanje nevronov je preprost način regularizacije učenja, ki ne zahteva računsko zahtevnih operacij, zato je pogosto uporabljen v nevronskih mrežah.

3.2 You Only Look Once

YOLO je model za detekcijo objektov in je bil javnosti predstavljen leta 2015 v [28]. Za razliko od takrat zelo popularnega modela R-CNN [8], YOLO ne potrebuje dveh ločenih nevronskih mrež za izločanje potencialnih kandidatov in klasifikacijo, ampak obe fazi izvaja ena mreža. Prednost mreže YOLO je zato hitrost detekcije, po navedbah članka [28] s 45-imi sličicami na sekundo (grafična procesna enota Titan X). Poleg tega se YOLO uči na celotni sliki in ne le na potencialnih kandidatih ospredja, zato bolje oceni kontekst objekta, ki ga detektira. Vendar pa ima YOLO v primerjavi z ostalimi modeli za detekcijo objektov slabšo natančnost, sploh na majhnih predmetih ter se okna prilagajajo manj točno.

3.2.1 Detekcija

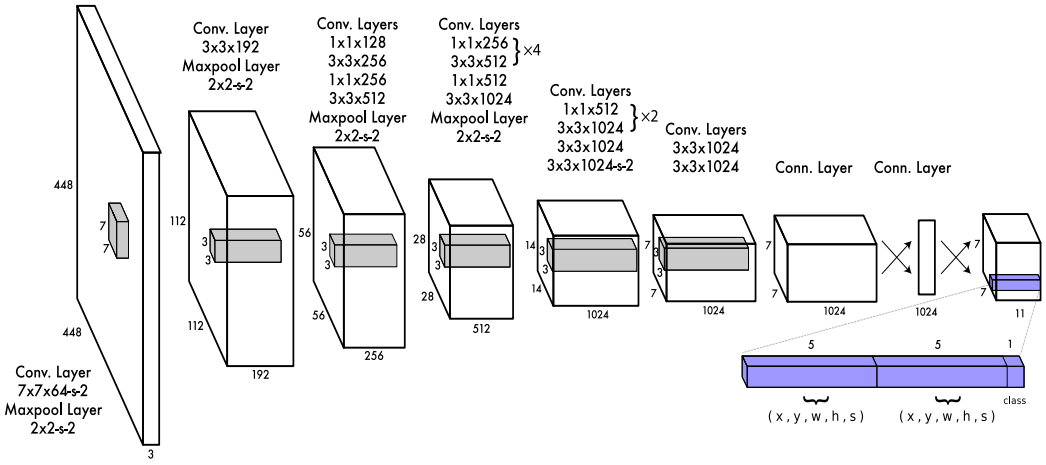
Učenje in detekcija potekata tako, da se slika razdeli na SxS mrežo. Velja, da če center objekta spada v polje, je to odgovorno za njegovo detekcijo. Vsako polje mreže napove B omejenih pravokotnikov, pri čemer je napoved sestavljena iz koordinat centra pravokotnika, njegove širine in višine, zaupanje v napoved ter verjetnosti pripadnosti tega objekta razredom, ki jih zaznavamo. Vrednost zaupanja izraža, koliko je model prepričan, da se na tem mestu nahaja dani objekt in natančnost omejenega pravokotnika. Če v nekem polju ni objekta, mora biti vrednost zaupanja čim manjša, drugače pa enaka IOU med napovedanim pravokotnikom in dejanskim pravokotnikom, ki oriše objekt. Potek detekcije objektov se nahaja na Sliki 8.



Slika 8: Detekcija objektov [28, Slika 2]

3.2.2 Arhitektura nevronske mreže YOLO

Model YOLO je implementiran kot konvolucijska nevronska mreža, sestavljena iz konvolucijskih plasti, ki izločijo značilke, ter polnopovezanih plasti, ki napovejo značilnosti detekcij. Pri zgradbi mreže (različica arhitekture, ki smo jo uporabili je narisana na Sliki 9) so se znanstveniki, ki so zasnovali YOLO, zgledovali po modelu GoogleNet [36]. Sestavljena je iz 24 konvolucijskih nivojev, ki jim sledijo dva polnopovezana nivoja. V naši implementaciji smo dodali še normalizacijo skupin, ker se je v naslednji različici modela YOLO izkazalo, da se natančnost detekcije ob tem dodatku izboljša za 2% [29]. Poleg tega smo število filtrov v prvi polnopovezani plasti znižali iz 4096 na 1024, saj je naš problem manj kompleksen. Tako smo pridobili na hitrosti inference.



Slika 9: Arhitektura naše spremenjene mreže YOLO - izhod je tenzor dolžine $S \cdot S \cdot (B \cdot 5 + C)$, pri čemer je bilo v našem primeru, ko smo detektirali samo en razred (roka), število razredov C ena

3.2.3 Izgubna funkcija

Model YOLO uporablja sestavljeno izgubno funkcijo, ki je definirana kot:

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2)] +$$

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2)] +$$

$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_i)^+$$

$$\lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_i) +$$

$$\sum_{i=0}^{S^2} \mathbb{1}_{i}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2$$

in je zgrajena iz petih komponent:

1. napaka koordinat centra omejenega pravokotnika,

2. napaka dimenzij omejenega pravokotnika,

3. napaka prisotnosti objekta, ki je najmanjša, ko je napovedana natančnost detekcije enaka IOU med napovedjo in resničnim omejenim pravokotnikom, če velja da celica vsebuje objekt,

4. napaka odsotnosti objekta, ki je najmanjša, ko je napovedana natančnost detekcije enaka 0 če velja, da celica ne vsebuje objekta in

5. napaka klasifikacije, ki je bila v našem primeru vedno nič, ker smo učili samo en razred.

Ker v času učenja in detekcije večina celic ne vsebuje objekta in bi bila napovedana natančnost detekcije zanje blizu nič, bi potencialno lahko prišlo do nestabilnosti modela ter zato do prezgodnje divergence. Ta problem uravnavata konstanti λ_{coord} in λ_{noobj}.

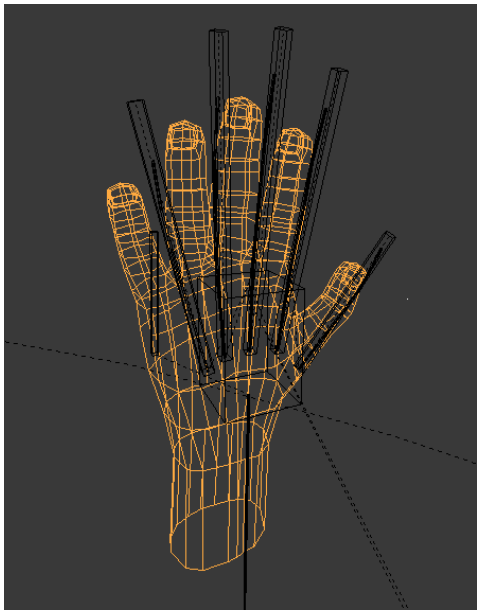
3.3 Generiranje sintetičnih podatkov

Umetnega generiranja podatkov smo se lotili v dveh fazah. V prvi fazi smo zajemali slike tridimenzionalnega modela roke, ki smo jih v drugem koraku združevali z ozadji, in tako ustvarili učno množico slik.

3.3.1 Generiranje slik dlani

Za generiranje sintetičnih podatkov smo uporabili tridimenzionalni model roke, ki smo ga naključno premikali in zajemali učne slike. Model je realističen in ga sestavlja 2590 poligonov ter 59 kosti. Ogrodje modela roke je prikazano na Sliki 10. Proces generiranja poz smo implementirali s skripto v orodju Blender[1]. Vsak prst se je z kombinacijo skaliranja in rotacij skrčil za naključen kot, poleg tega pa se je celotna dlan še naključno zarotirala.

[1]https://www.blender.org/



Slika 10: Tridimenzionalni model roke, ki smo ga uporabili za generiranje učnih slik

Za vsako tako pridobljeno pozo se je generiralo v najprej določeno število slik. Za vzpostavitev raznolikosti in čim večje fotorealističnosti smo naredili naslednje:

- prsti in zapestje roke so se med zajemanjem premikali za manjše kote,
- v prostor smo postavili dve luči, prostorsko in točkovno usmerjeno, katerih lokaciji in intenziteti sta se naključno spreminjali,
- da bi dosegli bolj realističen izgled kože, smo uporabili podpovršno sipanje (ang. *subsurface scattering*) [15], s katerim se modelira razpršitev svetlobe ob dotiku s površino roke (primerjava med uporabo in neuporabo tega mehanizma se nahaja na Sliki 11) in
- ustvarili smo tridimenzionalni model rokava majice, ki smo ga uporabili na delu zajetih slik ter katerega barva je bila izbrana naključno.

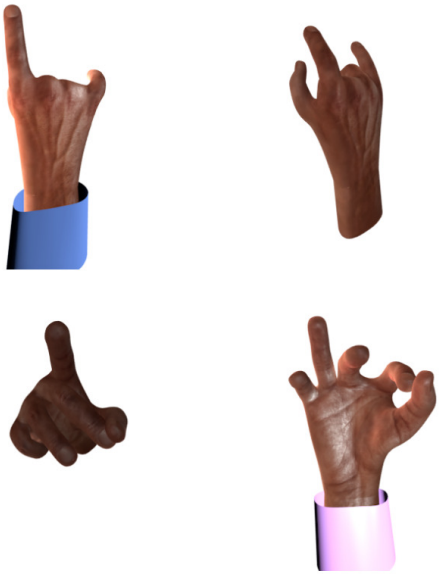
Slika 11: Primerjava uporabe (levo) ali neuporabe (desno) podpovršnega sipanja

Ob vsakem zajemu slike modela roke sta se shranili dve različici na prosojnem ozadju - ena je vsebovala celoten model, druga pa model brez zapestja. Slednjo smo v naslednjem koraku uporabili za ocenitev omejenega pravokotnika v katerem se nahaja dlan. Primeri tako generiranih slik rok so ponazorjeni na Sliki 12.

Proces generiranja učnih slik je avtomatiziran, uporabnik mora po generiranju poze le potrditi, ali je ta naravna in ali se prsti ne prekrivajo.

3.3.2 Združevanje slik dlani z ozadji

Po zajemu slik tridimenzionalnega modela roke je sledilo lepljenje prosojnih slik dlani na ozadja. Vsak generiran par slik rok smo augmentirali z naključno kombinacijo vertikalnega in horizontalnega preobrata, rotacij, obrezovanja, filtriranja z Gaussovim filtrom, ter sprememb v barvni in nasičenostni komponenti HSV barvnega prostora. Velikost slike dlani smo naključno spremenili ter jo prilepili na naključno izbrano koordinato v sliki ozadja (primeri učnih slik so na Sliki 13). Na podlagi alfa komponente smo v sliki roke brez



Slika 12: Primeri generiranih slik rok

zapestja poiskali omejen pravokotnik največje povezane komponente. Sliko, center in dimenzije tega omejenega pravokotnika smo shranili in jih uporabili pri učenju.

Slika 13: Primeri generiranih učnih slik

Poglavje 4

Eksperimentalno ovrednotenje

Da bi analizirali naš sistem za umetno generiranje slik rok in preizkusili naučeno detekcijo, smo naredili več eksperimentov. Za namen evaluacije smo anotirali dve realistični podatkovni zbirki ter na podoben način kot smo ustvarili učno množico slik, umetno generirali še 5000 slik za dodatno verifikacijo. Naučili smo več detektorjev z različnimi učnimi množicami, ki so jih sestavljale sintetične in realistične slike, ter primerjali, kako število učnih epoh in količine podatkov vpliva na učenje in natančnost detekcije. V naslednjih poglavjih je opisan postopek evaluacije in pridobljeni rezultati.

4.1 Umetno generiranje podatkov

Za kreiranje učne zbirke smo uporabili naš sistem za umetno generiranje slik rok in ustvarili 200 različnih poz, za vsako pozo 10 različic. V naslednjem koraku smo jih prilepili na ozadja, ki smo jih pridobili v iskalniku slik Google Images[1] z iskanjem različnih tem (narava, živali, ljudje, itd.). Slike smo pregledali, da se slučajno na kakšni izmed njih ne bi pojavile človeške dlani. Za kreiranje učne množice slik smo uporabili približno 7000 tako pridobljenih ozadij. Poleg tega smo za namen dodatne verifikacije generirali še 100 novih poz, zajeli 10 slik vsake in poiskali ter pregledali še približno 2000 ozadij.

[1] https://images.google.com/

4.2 Uporabljene podatkovne zbirke

Evaluacijo smo izvedli na treh podatkovnih množicah. Prvo, ki jo bomo označevali z oznako T1, smo pridobili na internetu [20, 21]. Sestavljajo jo slike, zajete z monokularno barvno kamero, in podatki, zajeti z napravama Kinect in Leap Motion. Zaradi narave naše diplomske naloge smo uporabili samo barvne slike. Dlani na slikah smo anotirali z orodjem labelImg[2]. Tako smo pridobili podatkovno množico s 1400 slikami (14 oseb, ki kaže 10 različnih gest, vsako po desetkrat) in pripadajočimi informacijami o omejenih pravokotnih, ki označujejo lokacije rok.

Ker so preostale roke na slikah ali zelo majhne, ali pa deloma prekrite, smo predpostavili, da model detektira le levo roko. Preostale roke na slikah smo v procesu evaluacije ignorirali, ker je bila možnost, da bi te roke naš detektor zaznal, zelo majhna. Primerov, kjer je slika vsebovala več rok je bilo malo, vendar pa so bili prisotni, eden izmed njih se nahaja na Sliki 14.

Slika 14: Primer slike, na kateri se nahajata dve dlani

Drugo podatkovno množico, ki jo bomo označevali z oznako T2, smo prav tako pridobili na internetu [5]. Sestavljena je iz 5440 barvnih in enako pripadajočih globinskih slik (10 oseb, ki kaže 34 različnih gest, vsako po

[2] `https://github.com/tzutalin/labelImg`

šestnajstkrat). Zaradi njene obsežnosti smo uporabili le del (10 gest, ki predstavljajo števila od 0 do 9 v ameriškem znakovnem jeziku) in teh 1600 slik, kot pri prejšnji podatkovni zbirki, ročno anotirali.

Ker je vhod v učni model slika velikosti $448 \cdot 448$ slikovnih elementov in smo želeli ohraniti razmerje stranic, smo slike obeh realističnih podatkovnih zbirk ob straneh obrezali, da smo dobili slike kvadratne oblike. To smo lahko naredili, saj se na nobeni sliki dlan ne nahaja ob robu slike, ampak so približno centralizirane.

Tretjo podatkovno zbirko, označevali jo bomo z oznako T3, smo generirali sami za namen dodatne analize na slikah, ki so podobne učnim. Sestavljena je iz 5000 slik, ki smo jih dobili tako, da smo ustvarili 100 novih gest in jih združili s 2000 ozadji, ki niso bila uporabljena v učnem procesu.

4.3 Učenje

Učenje smo izvajali na grafični kartici GeForce GTX 980 in je trajalo dobrih 24 ur. Uporabili smo optimizacijski algoritem Adam [13], velikost učnih skupin je bila 5.

Za učenje prvega učnega modela M1 smo uporabili izključno umetno generirane slike. Izvedli smo več učenj in testirali kako količina podatkov in število učnih epoh vplivata na natančnost detektorja. Učili smo največ 300000 korakov in vsakih 20000 korakov model shranili za evaluacijo. Uporabili smo naslednji učni režim:

- prvih 20000 korakov smo učili z učno stopnjo 0,00005,
- naslednjih 60000 korakov smo učili z učno stopnjo 0,00001,
- naslednjih 20000 korakov smo učili z učno stopnjo 0,000005,
- naslednjih 50000 korakov smo učili z učno stopnjo 0,000001,

- naslednjih 50000 korakov smo učili z učno stopnjo 0,0000005,
- zadnjih 100000 korakov smo učili z učno stopnjo 0,0000001.

Osnova drugega učnega modela M2 je bil najboljši model naučen v prejšnjem koraku, ki pa smo ga dodatno učili 10000 korakov na 100-ih slikah iz T1 (2 osebi, ki kažeta 5 gest vsako po desetkrat). Uporabili smo enako učno stopnjo kot v zadnjih korakih učenja na izključno sintetičnih podatkih.

Tretji učni model M3 smo učili samo na 100 slikah omenjenih v prejšnji točki, četrti učni model M4 pa na 80% naključno izbranih slik iz T1. Za učenje obeh učnih modelov smo uporabili enak režim učnih stopenj, kot je omenjen pri učenem modelu M1.

4.4 Evaluacijski protokol

V evaluacijski fazi smo točnost lokalizacije ocenili s tremi merami. Prva je natančnost, ki je definirana kot razmerje med številom pravilno označenih in vseh označenih omejenih pravokotnikov. Druga mera je priklic, ki je definiran kot razmerje med številom pravilno označenih in vseh omejenih pravokotnikov, ki bi morali biti označeni. Tretja uporabljena merila, ki združuje natančnost in priklic pa je F-mera in je definirana kot:

$$F = \frac{2 \cdot natančnost \cdot priklic}{natančnost + priklic}.$$

Za lažjo predstavo in oceno optimalnih parametrov detekcije smo izrisali PR krivulje, ki opisujejo razmerje med natančnostjo in priklicem pri različnih mejah sprejemanja zaznav rok.

Testiranje je potekalo v več fazah.

1. Sliko smo zmanjšali z originalne velikosti na velikost $448 \cdot 448$ slikovnih elementov, jo normalizirali (delili z 255) in prebrali pravilne podatke o

lokaciji in dimenziji omejenih pravokotnikov.

2. Izvedli smo detekcijo, katere rezultat je bila množica omejenih pravokotnikov (koordinate centra, dimezije in prepričanost v pravilnost zaznave).

3. Z namenom zmanjšanja števila ponavljajočih se omejenih pravokotnikov, smo le-te obdelali z metodo dušenja nemaksimalnih vrednosti, ki je potekala tako:

   - najprej smo vsak omejeni pravokotnik primerjali z obstoječimi skupinami in če je bilo njegovo prekrivanje s katerimkoli omejenim pravokotnikom iz te skupine večje kot 10%, smo ga vključili v to skupino,
   - če je bila njegova prepričanost v pravilnost večja kot maksimalna prepričanost v tej skupini, je postal nosilec in
   - če se omejeni pravokotnik ni prekrival z nobeno skupino, smo ustvarili novo, katere nosilec je postal.

4. Po izvedbi dušenja nemaksimalnih vrednosti smo pregledali nosilca vsake skupine. Če je bila njegova prepričanost v zaznavo večja od optimalne meje, ki smo jo pridobili z izrisom PR krivulje, smo ga sprejeli kot predlagani omejeni pravokotnik, sicer pa smo skupino zavrgli.

5. Vsako detekcijo smo ovrednotili tako, da smo izračunali IOU s pravilnim omejenim pravokotnikom. Če je bil ta večji od 50%, smo prišteli ena k številu pravilnih zaznav (ang. *true positive*), če pa je bil IOU manjši ali pa je bil ta objekt že zaznan, smo prišteli ena k številu napačnih zaznav, ki so bile označene kot pravilne (ang. *false positive*).

6. Če objekt ni bil nikoli zaznan pravilno, smo dodali ena k številu napačno nedetektiranih objektov (ang. *false negative*).

7. Glede na število glasov v treh skupinah, omenjenih v prejšnji točki, smo izračunali natančnost, priklic in F-mero.

4.5 Rezultati

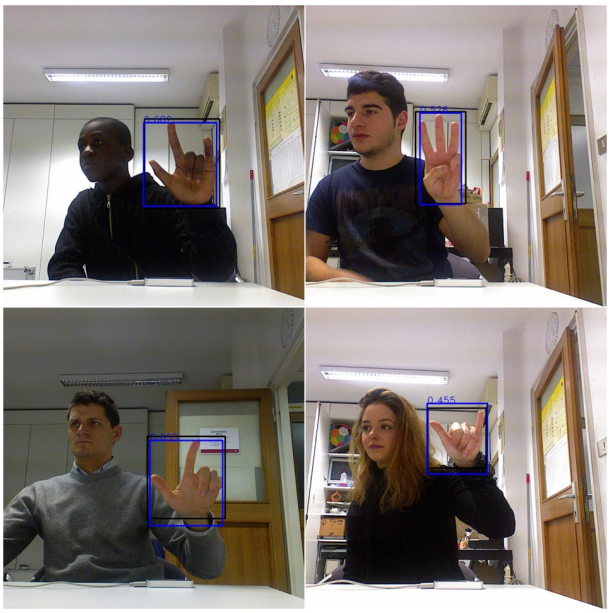
V nadaljevanju so predstavljeni rezultati, ki so jih naučeni modeli dosegli na treh testnih množicah slik.

4.5.1 Rezultati modela, naučenega iz izključno sintetičnih slik

Po učenju smo opravili več eksperimentov, da bi videli, ali so prisotni vplivi na uspešnost detekcije glede na število učnih slik in epoh. Pokazalo se je, da najboljše rezultate učenja iz izključno sintetičnih podatkov dosežemo, če uporabimo 50000 učnih slik, ki jih učimo 280000 korakov (ker smo uporabili učne podmnožice velikosti 5, to pomeni, da smo učili 28 epoh). Izkazalo se je, da je za učenje nevronske mreže zelo pomemben faktor kontekst. Najprej smo učili na generiranih slikah rok, ki so bilo odrezane nad zapestjem (primeri takšnih slik se nahajajo na Sliki 15), vendar pa smo pri primerjavi z modelom, ki se je učil iz slik rok, ki vsebujejo zapestje ugotovili, da je v drugem primeru naučeni detektor boljši. Potrdil se je tudi znan problem slabega prilagajanja omejenih pravokotnikov, ki jih oceni model YOLO. Velikokrat se je namreč zgodilo, da je model detektiral pravilen objekt, vendar pa je bila zaznava označena kot napačna, ker je bil IOU premajhen.
Pri testiranju na T1 smo dosegli precej visoko F-mero. Ker je ta zbirka slik dokaj enostavna (dlani so vedno obrnjene proti kameri, osvetlitev je dobra in resolucija slik visoka), je to tudi pričakovan rezultat. Razlogi za napačne detekcije so predvsem v preslabem ujemanju omejenih pravokotnikov. To se tudi kaže na PR krivulji (padec v natančnosti tudi, ko je priklic majhen), ker je detektiranih veliko omejenih pravokotnikov, ki imajo visoko prepričanost, vendar pa jih označimo kot napačne zaradi nizkega IOU. Primeri pravilnih detekcij se nahajajo na Sliki 16, primeri napačnih in razlogi zanje pa na Sliki 17.

Podatkovna zbirka T2 je veliko bolj zahtevna zaradi nižje resolucije slik,

Slika 15: Primeri prvotno generiranih slik rok, ki so bile odrezane nad zapestjem



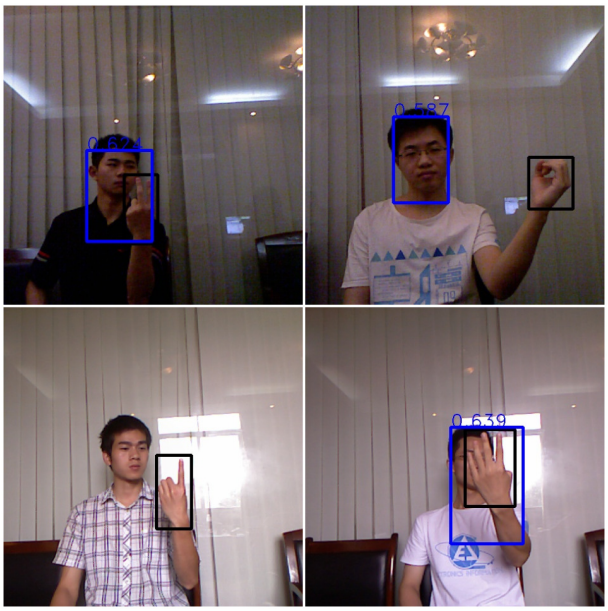
Slika 16: Primeri uspešno detektiranih dlani v ekperimentu M1_T1

Slika 17: Primeri napačnih detekcij dlani iz eksperimenta M1_T1, zaradi zameglitve, ki je posledica premika (levo zgoraj), slabe osvetlitve (levo spodaj), napačno detektiranega obraza (desno zgoraj) in prenizkega IOU (desno spodaj)

manjših rok v primerjavi s testno množico T1 (znana pomanjkljivost mreže YOLO je občutljivost na majhne predmete), večje raznolikosti slik v rotacijah zapestij rok in osvetlitvi ter pogostejši prisotnosti zameglitve zaradi gibanja. To so tudi razlogi, da je najboljša dosežena F-mera precej nižja kot pri M1_T1. Primeri pravilno zaznanih dlani se nahajajo na Sliki 18, primeri napačnih zaznav in razlogi zanje pa na Sliki 19. Ob izrisu PR krivulje in pregledu napačnih detekcij se je izkazalo, da je relativno nizka F-mera posledica tudi veliko napačno pozitivnih detekcij na obrazih. Ta problem smo omilili z uporabo večjega števila ozadij, ki vsebujejo človeške obraze v učni množici, vendar pa je problem ostal vseeno prisoten.

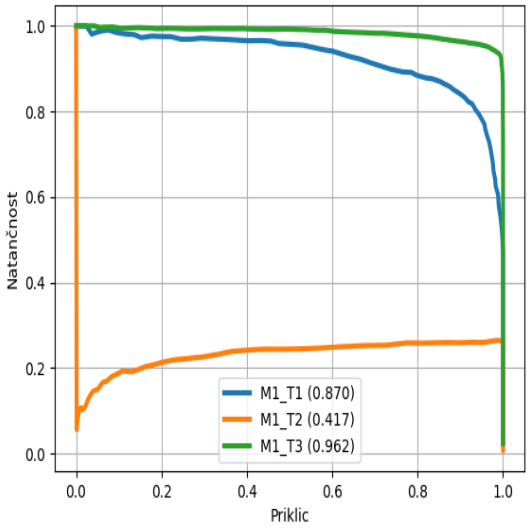
Z namenom boljšega natančnejšega vpogleda v točnost detekcije na slikah, ki so podobne učni množici, smo izvedli analizo še na podatkovni zbirki T3,

Slika 18: Primeri uspešno detektiranih dlani pri M1_T2



Slika 19: Primeri napačnih detekcij dlani pri M1_T2, zaradi razširitve omejenega pravokotnika na obraz (levo zgoraj), osvetlitve od zadaj (levo spodaj), napačno detektiranega obraza (desno zgoraj) in prenizkega IOU (desno spodaj)

ki vsebuje izključno umetno generirane slike rok. Dosežena F-mera je bila pričakovano visoka. Ko smo preučili napačne detekcije smo ugotovili, da so te pogosto posledica spregleda manjših rok. Glede na to, da dosežena F-mera ni 1, lahko sklepamo, da se model ni povsem naučil variance, ki je vsebovana v učni množici. Ker se je po določenem številu korakov natančnost naučenega modela na realističnih slikah začela zmanjševati in se je model začel učiti značilnosti roke, ki niso enake realnosti, nadaljnje učenje ni bilo več smiselno.
PR krivulje in dosežene najboljše F-mere prvega naučenega modela se nahajajo na Sliki 20.

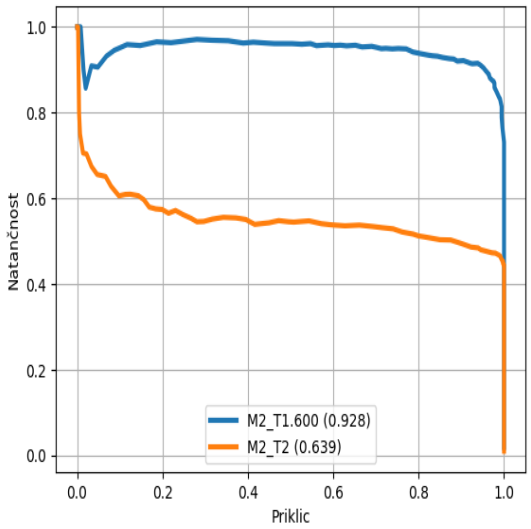


Slika 20: PR krivulje naučenega modela M1

4.5.2 Rezultati modela, naučenega iz kombinacije sintetičnih in realističnih podatkov

V naslednjem koraku smo model M1 dodatno učili (ang. *finetune*) na majhni množici realističnih slik (100 slik) iz T1. Za evaluacijo tega naučenega modela, smo uporabili 600 slik iz T1 (za boljše razločevanje to podatkovno

množico imenujmo T1.600), tako da v učnem procesu niso bile uporabljene slike istih oseb ali gest kot v evaluacijskem procesu in T2. Na obeh podatkovnih zbirkah so se rezultati detekcije izboljšali, kar kaže na to, da se v fazi učenja na izključno sintetičnih slikah učni model ne nauči določenih pomembnih značilk, ki so vsebovane v realnih primerih. Iz rezultatov in krivulje M2_T2, izrisane na Sliki 21 lahko opazimo, da so primeri, ko se kot pozitivna zaznava detektira obraz, še vedno prisotni, vendar pa jih je manj.

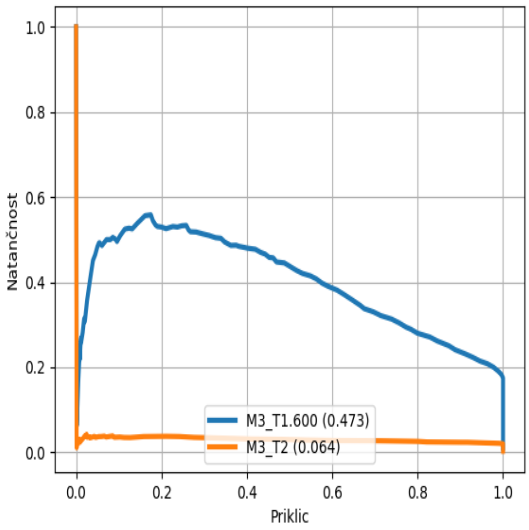


Slika 21: PR krivulji naučenega modela M2

4.5.3 Rezultati modela, naučenega iz 100 realističnih slik

Tretji učni model, M3, smo učili na 100 slikah, ki smo jih za dodatno učenje uporabili pri M2, zato, da bi preverili, ali je generiranje sintetičnih slik v primerjavi z zajemanjem manjše količine realističnih slik sploh smiselno. Čas, ki ga porabimo za umetno generiranje slik, je primerljiv oziroma celo krajši kot čas, ki bi ga porabili za zajem in anotacijo 100 realističnih slik. Izkazalo se je, da je F-mera modela M3 na obeh realističnih podatkovnih zbirkah

nižja (prikazano na Sliki 22) tudi od F-mere, ki jo doseže M1. Zato lahko zaključimo, da je umetno generiranje slik smiselno, saj nam prihrani veliko časa in človeške delovne sile.

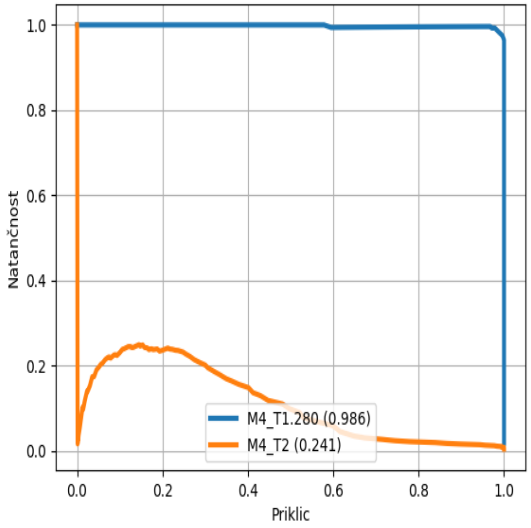


Slika 22: PR krivulji naučenega modela M3

4.5.4 Rezultati modela, naučenega iz naključnega dela prve podatkovne zbirke

Četrti učni model, M4, smo učili na naključno izbranih 80% slik iz T1 in testirali na preostalih 20% (to testno množico poimenujmo T1.280). Dosežena F-mera je bila pričakovano blizu 1, saj je deset primerov vsake geste, ki jo pokaže ista oseba zelo podobnih (manjši premiki roke, spremembe ozadja). Model M4 smo naučili, da bi dobili občutek, kolikšna je maksimalna F-mera, ki jo lahko naučimo na podlagi dane podatkovne zbirke s konvolucijsko nevronsko mrežo, ki smo jo uporabili. Na tretji podatkovni zbirki je model M4 dosegel nizko F-mero tudi zato, ker veliko gest, ki so bile pokazane na teh slikah, ni bilo v učnem procesu. Izrisani PR krivulji se nahajata na Sliki 23. Pokazalo se je, da tudi veliko realnih slik ni dovolj za prenos znanja na drugo

podatkovno množico in je v tem primeru bolje uporabiti umetno generirane podatke. Obenem pa se lahko (kot se je pokazalo pri eksperimentu M2_T2) z dodajanjem malo realnih slik vidno izboljšajo rezultati, tudi če so te slike iz druge podatkovne množice.



Slika 23: PR krivulji naučenega modela M4

Poglavje 5

Zaključek

V diplomskem delu smo učili detektor dlani z uporabo konvolucijskih nevronskih mrež. Uporabljene podatke smo pri tem generirali umetno. Sintetično učno množico slik je na tem primeru možno ustvariti precej enostavno z uporabo tridimenzionalnega modela roke, saj so roke posameznikov med seboj podobne, poleg tega pa tudi sama roka nima veliko stopenj svobode. Eden izmed ciljev je bil narediti sistem za avtomatizirano generiranje učnih slik, ki bi nadomestil alternativno dolgotrajno zajemanje in anotacijo.

Primerjali smo točnost zaznave pri uporabi izključno realističnih slik, izključno sintetičnih slik ali pri kombinaciji obojega. Predvsem smo se usmerili na področje komunikacije človek računalnik in temu tudi prilagodili učne podatke. Naš naučeni model deluje v omejenem scenariju, kjer so roke vsaj deloma usmerjene proti kameri, ne bo pa deloval v splošnem, na primer, če so roke majhne ali jih deloma zakriva kakšen predmet. Ker je zaželeno, da komunikacija poteka v realnem času, smo uporabili mrežo YOLO, ki je sicer manj točna, a hitrejša. Izkazalo se je, da lahko dosežemo dobre rezultate zaznave tudi z izključno umetno generiranimi slikami, če je le raznolikost učnih podatkov dovolj visoka in natančno prikazujejo realnost. Da bi to dosegli, smo uporabili več mehanizmov (raznolike poze, naključno spreminjanje osvetlitve, različne augmentacije, podpovršno sipanje in uporabo konteksta).

Pokazalo se je, da z dodajanjem manjšega deleža realističnih slik dosežemo boljše rezultate kot če uporabimo izključno sintetične podatke. Kljub temu sklepamo, da to v primeru podrobnejše analize razlik značilk, ki se jih model nauči v obeh primerih, ne bi bilo potrebno. Ena izmed večjih pomanjkljivosti, ki smo jo sicer želeli odpraviti s podaljšanjem zapestja in dodajanjem rokavov, je to, da so bile roke na slikah naključno postavljene, kar pa v realnosti ni res. Pri učenju nevronskih mrež je namreč tako, kot pri delovanju človeških možganov, zelo pomemben kontekst.

Diplomsko delo ponuja kar nekaj iztočnic za nadaljnje raziskave, ki bi omogočile boljše generiranje sintetičnih podatkov in spodbujanje nevronske mreže k učenju značilk, ki natančneje predstavljajo svet okoli nas. Učenje bi si želeli avtomatizirati tako, da bi ustvarili povratno zanko, ki bi z iskanjem razlogov za napačne in manjkajoče detekcije ter dodajanjem takšnih težjih primerov omogočala izboljšanje naslednje ponovitve učenja (ang. *hard-negative / positive mining*). Smiselno bi bilo preizkusiti tudi druge mreže za detekcijo. Poleg tega bi lahko detekciji dodali še razpoznavanje gest in tako rezultat diplomskega dela razširili v sistem, ki bi omogočal komunikacijo človek računalnik v realnem času.

Literatura

[1] J. Baek, J. Kim, C. Yoon, D. Kim, and E. Kim. Part-based hand detection using hog. 23:551–557, December 2013.

[2] Y. Cui, D. L. Swets, and J. J. Weng. Learning-based hand sign recognition using shoslif-m. In *Proceedings of IEEE International Conference on Computer Vision*, pages 631–636, June 1995.

[3] N. H. Dardas and N. D. Georganas. Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. *IEEE Transactions on Instrumentation and Measurement*, 60:3592–3607, 2011.

[4] A. Dobnikar. *Nevronske mreže: teorija in aplikacije*. Didakta, 1990.

[5] B. Feng, F. He, X. Wang, Y. Wu, H. Wang, S. Yi, and W. Liu. Depth-projection-map-based bag of contour fragments for robust hand gesture recognition. *IEEE Transactions on Human-Machine Systems*, 47(4):511–523, August 2017.

[6] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. 1994.

[7] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 580–587, Washington, DC, USA, 2014. IEEE Computer Society.

[9] J. Guo, J. Cheng, J. Pang, and Y. Guo. Real-time hand detection based on multi-stage hog-svm classifier. In *2013 IEEE International Conference on Image Processing*, pages 4108–4111, September 2013.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[11] S. N. Karishma and V. Lathasree. Fusion of skin color detection and background subtraction for hand gesture segmentation. 2014.

[12] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun. Real time hand pose estimation using depth sensors. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1228–1234, Nov 2011.

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[14] M. Kolsch and M. Turk. Robust hand detection. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pages 614–619, May 2004.

[15] A. Krishnaswamy and G. V. G. Baranoski. A Biophysically-Based Spectral Model of Light Interaction with Human Skin. *Computer Graphics Forum*, 2004.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges,

L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[17] A. Kurakin, Z. Zhang, and Z. Liu. A real time system for dynamic hand gesture recognition with a depth sensor. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 1975–1979, August 2012.

[18] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989]*, pages 396–404, 1989.

[19] C. Li and K. M. Kitani. Pixel-level hand detection in ego-centric videos. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3570–3577, June 2013.

[20] G. Marin, F. Dominio, and P. Zanuttigh. Hand gesture recognition with leap motion and kinect devices. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 1565–1569, October 2014.

[21] G. Marin, F. Dominio, and P. Zanuttigh. Hand gesture recognition with jointly calibrated leap motion and depth sensor. *Multimedia Tools Appl.*, 75(22):14991–15015, November 2016.

[22] X. Meng, J. Lin, and Y. Ding. An extended hog model: Schog for human hand detection. In *2012 International Conference on Systems and Informatics (ICSAI2012)*, pages 2593–2596, May 2012.

[23] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt. Ganerated hands for real-time 3d hand tracking from monocular RGB. *CoRR*, abs/1712.01057, 2017.

[24] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt. Real-time hand tracking under occlusion from an egocentric RGB-D sensor. *CoRR*, abs/1704.02201, 2017.

[25] S. O., R. Mallipeddi, and M. Lee. Real time hand gesture recognition using random forest and linear discriminant analysis. In *Proceedings of the 3rd International Conference on Human-Agent Interaction*, HAI '15, pages 279–282, New York, NY, USA, 2015. ACM.

[26] P. Premaratne. *Human Computer Interaction Using Hand Gestures*. Springer Publishing Company, Incorporated, 2014.

[27] M. A. Rahman, I. K. Edy Purnama, and M. H. Purnomo. Simple method of human skin detection using hsv and ycbcr color spaces. In *2014 International Conference on Intelligent Autonomous Agents, Networks and Systems*, pages 58–61, Aug 2014.

[28] J. Redmon, S. Kumar Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[29] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016.

[30] J. M. Rehg and T. Kanade. Digiteyes: Vision-based hand tracking for human-computer interaction. In *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 16–22, November 1994.

[31] E. Richardson, M. Sela, and R. Kimmel. 3d face reconstruction by learning from synthetic data. *CoRR*, abs/1609.04387, 2016.

[32] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *European Conference on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016.

[33] D. B. Rubin. Discussion statistical disclosure limitation.

[34] J. Segen and S. Kumar. Gesture vr: Vision-based 3d hand interace for spatial interaction. pages 455–464, January 1998.

[35] B. Sun and K. Saenko. From virtual to reality: Fast adaptation of virtual object detectors to real domains. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[37] R. Szeliski. Computer vision: Algorithms and applications, 2010.

[38] C. Tsai, Y. Tsai, S. Hsu, and Y. Wu. Synthetic training of deep cnn for 3d hand gesture identification. *2017 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO)*, pages 165–170, 2017.

[39] W. Wang and J. Pan. Hand segmentation using skin color and background information. In *2012 International Conference on Machine Learning and Cybernetics*, volume 4, pages 1487–1492, July 2012.

[40] C. Zimmermann and T. Brox. Learning to estimate 3d hand pose from single rgb images. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.