

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
PROGRAMŲ SISTEMŲ BAKALAURO STUDIJŲ PROGRAMA

Krepšinio taisyklių pažeidimo automatinis nustatymas

Recognizing violations of basketball rules using computer vision

Bakalauro baigiamasis darbas

Atliko:	Lukas Cedronas	(parašas)
Darbo vadovas:	partn. prof., dr. Vytautas Ašeris	(parašas)
Darbo recenzentas:	lekt. Donatas Kimutis	(parašas)

Vilnius – 2021

Santrauka

Glaustai aprašomas darbo turinys: pristatoma nagrinėta problema ir padarytos išvados. Santraukos apimtis ne didesnė nei 0,5 puslapio. Santraukų gale nurodomi darbo raktiniai žodžiai.

Raktiniai žodžiai: raktinis žodis 1, raktinis žodis 2, raktinis žodis 3, raktinis žodis 4, raktinis žodis 5

Summary

Santrauka anglų kalba. Santraukos apimtis ne didesnė nei 0,5 puslapio.

Keywords: keyword 1, keyword 2, keyword 3, keyword 4, keyword 5

TURINYS

ĮVADAS	4
1. NAUDOTI ĮRANKIAI	5
2. VAIZDO MEDŽIAGOS PARUOŠIMAS	6
3. VAIZDO ATPAŽINIMO METODAI	7
3.1. Spalvinis atpažinimas	7
3.1.1. Segmentavimas	7
3.1.2. Morfolginės transformacijos	8
3.1.3. Kamuolio kontūrų radimas	9
3.2. Atpažinimas remiantis skirtumais	10
3.2.1. Fono pašalinimas	10
3.2.2. Judesio atpažinimas	10
3.3. Žmogaus kūno dalių atpažinimas	11
3.4. Kamuolio atpažinimas	12
REZULTATAI IR IŠVADOS	14
LITERATŪRA	15
SANTRUMPOS	16
PRIEDAI	16
1 priedas. Neuroninio tinklo struktūra	17
2 priedas. Eksperimentinio palyginimo rezultatai	18

Įvadas

Įvade nurodomas darbo tikslas ir uždaviniai, kuriais bus įgyvendinamas tikslas, aprašomas temos aktualumas, apibrėžiamas tiriamasis objektas akcentuojant neapibrėžtumą, kuris bus išspręstas darbe, aptiriamos teorinės darbo prielaidos bei metodika, apibūdinami su tema susiję literatūros ar kitokie šaltiniai, temos analizės tvarka, darbo atlikimo aplinkybės, pateikiama žinių apie naudojamus instrumentus (programas ir kt., jei darbe yra eksperimentinė dalis). Darbo įvadas neturi būti dėstymo santrauka. Įvado apimtis 2 — 4 puslapiai.

1. Naudoti įrankiai

Kompiuterinės regos ir taisyklių pažeidimo atpažinimo algoritmams įgyvendinti buvo pasirinkta Python programavimo kalba. Python – interpretuojama, lengvai skaitoma kalba, puikiai tinkama įvairioms problemoms spręsti [Kuh12]. Dėl kalbos paprastumo ji dažnai naudojama kompiuterinės regos ir giliojo mokymo srityse, kadangi kalba leidžia susifokusuoti į abstrakcijas. Kadangi kalba yra plačiai naudojama, nesunku rasti daug pavyzdžių bei šaltinių.

Objektų atpažinimui buvo pasirinkta OpenCV biblioteka. Tai – nemokama biblioteka, plačiai naudojama spręsti kompiuterinės regos uždavinius, kurios fokusas – įgalinti kurti aplikacijas leidžiančias efektyviai spręsti kompiuterinės regos problemas realiu laiku [BK08].

Vaizdo medžiaga rinkta filmuojant Xiaomi Redmi 8 Pro kamera.

2. Vaizdo medžiagos paruošimas

Šio darbo metu sukurtai programinei įrangai paruošta vaizdo medžiaga, kurioje žaidėjas atlieka įvairius judesius su krepšinio kamuoliu. Siekiant supaprastinti vaizdo atpažinimo algoritmą, medžiagai keliami reikalavimai yra: aiškiai matomas žaidėjas, telpantis į kadrą ir užimantis nemažą dalį vaizdo. Vaizdo atpažinimui pagal spalvą algoritmui žaidėjas taip pat privalo dėvėti skirtingų spalvų pirštines, batus bei mušinėti atskiros spalvos kamuolį. Žaidėjas turi būti kiek galima labiau skirtis nuo fono, kad algoritmas veiktų kuo efektyviau, kadangi įvairių formų ir spalvų objektai fone pasunkina tikslų atpažinimą. Vaizdo medžiagoje žaidėjas mušinėja kamuolį ir atlieka įvairius judesius, dalis kurių pažeidžia taisykles, pavyzdžiui – pamušinėjęs kamuolį jį pasiima į rankas ir padaro keletą žingsnių. Programinė įranga turi atpažinti, jog tai – taisyklės pažeidimas. Vaizdo kokybė yra itin svarbi siekiant tikslių rezultatų, todėl buvo filmuojama kuo didesne raiška – Xiaomi Redmi Note 8 Pro kameros maksimali raiška yra 4K, 30 kadrų per sekundę.

3. Vaizdo atpažinimo metodai

Vaizdo atpažinimas kompiuteriu yra vis dar problema. Egzistuoja daug algoritmų, kurių pagalba galima atpažinti objektus. Sudetingesniems objektams dažnai yra naudojami giliuoju mokymu paremti metodai, šiame darbe išnagrinėti keli iš jų. Atpažinti objektą reiškia priskirti jam kažkokį kontekstą. Tačiau kartais užtenka tiesiog //@todo Krepšinyje atpažinimas kažką pavaryti apie state of the art

3.1. Spalvinis atpažinimas

Vienas iš elementariausių būdų išskirti objektus ir priskirti jiems kontekstą yra tada, kai žinome jų spalvą iš anksto //@todo rasti kuo paremti. Tuomet pagal spalvų reikšmes nesunkiai galime išskirti regionus, juos reprezentuojant vienetukų ir nuliukų matricomis, ir atlikti įvairias logines operacijas.

Tokio būdo trūkumas yra tai, jog norint panaudoti šį metodą atpažinti rankai ar kamuoliui, reikia iš anksto žinoti, kokios spalvos bus tie objektai. Negana to, labai sudėtinga išskirti atskiras kūno dalis naudojantis informacija apie odos spalvą, kadangi ne tik žaidėjo rankos, kojos bei veidas tikėtina bus tos pačios spalvos, bet galimai sutaps ir su kitų žaidėjų.

Naudojantis spalvomis limitacijos yra labai aiškos... Siekiant ištirti spalvinių metodų efektyvumą ir pritaikomumą, būtina naudoti tam tikros spalvos pirštines, batus bei kamuolį, tačiau realiaame pasaulyje žaidėjus priversti nešioti skirtingų spalvų batus, pirštines būtų itin nepraktiška. //@todo iškelti į išvadas ir pagrįsti Šie būdai tinkami tam tikrais išimtiniais atvejais. Šiame darbe taip pat bus nagrinėjami dar keli būdai, nepriklausantys nuo spalvų.

3.1.1. Segmentavimas

Segmentavimą kompiuterinėje regoje galima suprasti kaip radimą grupės panašių pikselių. Kai paveikslėliai segmentuojami pagal spalvą, panašumą galima išmatuoti pagal reikšmių skirtumą RGB arba HSV modeliuose. Tai – gan paprastas būdas išskirti dominantį regioną, kadangi užtenka apsibrėžti tam tikrą intervalą spalvų, paveikslėlyje, reprezentuotame trimatine matrica, ir atfiltruoti matricos reikšmes nepatenkančias į duotąjį intervalą. RGB modelis, spalvą koduojantis trimis reikšmėmis, nusakančiomis raudonumą, žalsvumą ir mėlynumą. Turi trūkumą - keičiantis apšvietimui nenusipėjamai gali pasikeisti spalvos RGB reikšmė, kas filtravimą padaro itin sudėtingą. Problemą išsprendžia HSV modelis, kurio modelis spalvą reprezentuoja atspalviu, ryškumu ir šviesumu. HSV modelis puikiai tinka spalvų atpažinimui, kadangi H (atspalvio) reikšmė mažai kinta paveikslėliuose atsiradus šešėliams [LBC02] ar miglotumui [WC15]. Atspalvis dažniausiai išlieka tas pats nepriklausomai nuo pašalinių efektų, todėl lengva nusistatyti reikšmių intervalą. Konvertavimas iš RGB ir HSV yra pakankamai trivialus, implementuotas cvtColor metodu.

Turint intervalą užtenka atfiltruoti visus pikselius, kurių reikšmė nepatenka į duotą intervalą. Tiems, kurie patenka OpenCV priskiria 1, o kurie ne - 0. Rezultate gauname išskirtą regioną kur žinome, kad tai - tam tikra dalis, kadangi tik ji turėjo tą spalvą. //@todo style



(a) Vaizdas prieš segmentavimo pagal pirštinės spalvą



(b) Vaizdas po segmentavimo pagal pirštinės spalvą

1 pav. Fono Aaaa.

Šiame darbe naudojamų spalvų režiai // @todo add spalvos . // @todo pagalvoti ar reikia <Pastebėta, jog lauke esančių paveikslėlių saturacija žymiai didesnė, nei kambaryje>

3.1.2. Morfologinės transformacijos

Atlikus segmentavimą pagal spalvą, rezultate dažnai lieka triukšmo ar bereikalingų artefaktų, kurie gali affectinti tolimesnį apdorojimą. @todo fix

Tokiais atvejais apsisvalyti nuo triukšmo gan dažnai naudojamos slenkstinės (*angl.* threshold) operacijos [Sze21, 112]. Apsibrėžus tam tikrą filtrą, su juo galima atlikti operacijas ant matricų. Šiame darbe naudotos operacijos yra erozija ir plėstis. Erozija naudojama tada, norima sumažinti turimų atskirų regionų paveikslėlyje plotus, nuardant kraštus. Jei objektai yra pakankamai maži, o erozijos operacija - stipri (?), tuomet taip pašalinamas triukšmas. Formaliai erozija buvo aprašyta 1980 metais K. Y. Yoo [HSZ87] kaip morfologinė operacija, sukombinuojanti dvi aibes naudojant vektorių atimtį aibių elementams. Matematiškai tai galima aprašyti tokia formule:

$$A \ominus B = \bigcap_{b \in B} (A)_{-b}. \quad (1)$$

1 lygtyje A žymi dominantį paveikslėlį, išreikštą matrica, o B - filtrą arba kitaip - struktūrinį elementą, su kuriuo A matricoje pakartotinai atliekamos sankirtos.

Erozijai priešinga operacija yra plėstis ?? Ši operacija leidžia padidinti išskirtų regionų plotą. Plėstis naudinga tuomet, kai po erozijos operacijos objektai praranda per daug ploto. Plėstį galima užrašyti šitaip:

$$A \oplus B = \bigcup_{b \in B} (A)_b. \quad (2)$$

Siekiant geriausių rezultatų šalinant triukšmą, pastebėta, jog plėstį geriausia vykdyti po erozijos, priešingu atveju triukšmas gali nebūti panaikintas.

OpenCV įgyvendina funkcijas erode ir dilate, kurios priima paveikslėlio matricą ir struktūrinių elementą. Taip pat egzistuoja funkcija morphologyEx, kuri, priklausomai nuo paduodamų parametrų keičia savo veikimą: su MORPH_OPEN atliekama plėstis po erozijos, su MORPH_CLOSE - atvirkščiai.

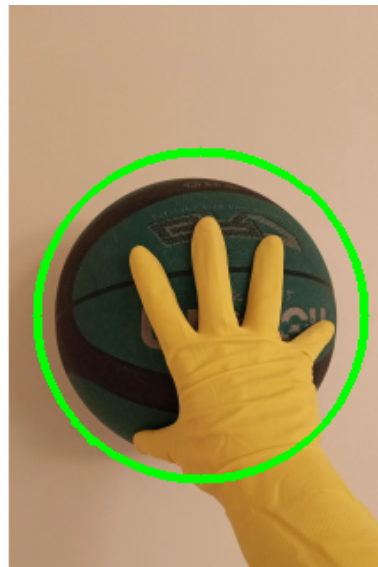
3.1.3. Kamuolio kontūrų radimas

//@todo pagalvoti ar prie spalvu turi eit Siekiant iš paveikslėlių išgauti daugiau kontekstualios informacijos, pavyzdžiui, ar kamuolys yra rankose, išsiskirti tarpusavy nesusikertančius regionus neužtenka. Tam, kad gauti šią informacijos dalį, reikia rasti, kada rankos ir kamuolio regionai susikerta, t.y. jų matricų sankirta yra netuščia. Tam reikia žinoti, kokią erdvę apima kamuolys - šios informacijos trūksta po segmentavimo pagal spalvą, jei kamuolys yra uždengtas, pavyzdžiui, rankos.

Vienas iš būdų šią problemą išspręsti - pabandyti atspėti, kokią sritį užima kamuolys randant jo kontūrus. Skritulių kontūrams rasti 1991 metais E. Welzl pasiūlė algoritmą, rekursiškai apskaičiuojantį tam tikrai taškų aibei mažiausią ją apimančią apskritimą [Wel91]. Šis algoritmas - tiesinio kompleksiskumo, jį įgyvendina OpenCV funkcija minEnclosingCircle().



(a) Kamuolio kontūrai prieš mažiausio aibę apimančio apskritimo radimo algoritmą



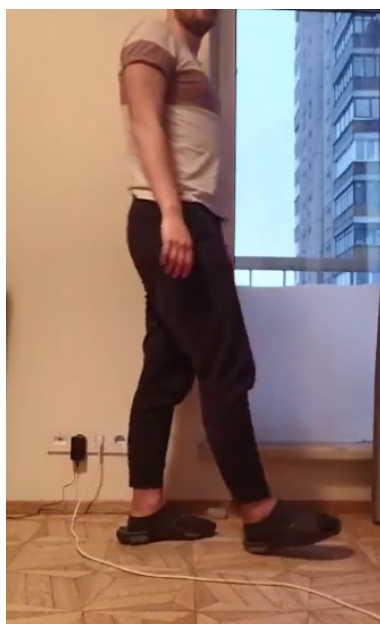
(b) Kamuolio kontūrai po mažiausio aibę apimančio apskritimo radimo algoritmo

2 pav. Fono Aaaa.

3.2. Atpažinimas remiantis skirtumais

3.2.1. Fono pašalinimas

Fono pašalinimas yra vienas iš esminių metodų kompiuterinėje regoje, naudojamas išskirti dominančius vaizdus ir pašalinti nereikalingus statiškus objektus iš fono. Pavyzdžiui, jei yra filmuojama lauke, fone besimatantys medžiai gali trukdyti tolimesniam atpažinimui, tad žinant, jog mus domina tik žaidėjas, pašalinius objektus galima pabandyti pašalinti. Vienas iš būdų yra iš anksto turėti fono paveiksluką ir apdorojant kitus kadrus jį išimti, bet tai dažnai nepasiteisina dėl to, kad fonas gali kisti – gali atsirasti šešėlių, pasikeisti apšvietimas, gali būti įvairaus pašalinio judėjimo, pavyzdžiui – linguojantys medžiai, vaikstantys sirgaliai ir pan. 2002 m. P. KaewTraKulPong pasiūlė algoritmą, išsprendžiantį šią problemą [KB02]. Algoritmas kiekvieną pikselį sumodeliuoja į Gauso skirstinį pagal tai, kiek pikselio spalva keičiasi bėgant laikui. Kuo mažiau pikseliai keičiasi, tuo didesnė tikimybė, kad jie priklauso fonui. Naudojantis šiuo atradimu galima sėkmingai atsikratyti fono, taip išskiriant mus dominantį vaizdą. Šis metodas ypač tinkamas atpažinti judėjimui – vietoje stovintis žaidėjas bus priskirtas fonui, tačiau jam judant, lengvai galima išskirti jo siluetą.



(a) Vaizdas prieš fono pašalinimą.



(b) Vaizdas po fono pašalinimo.

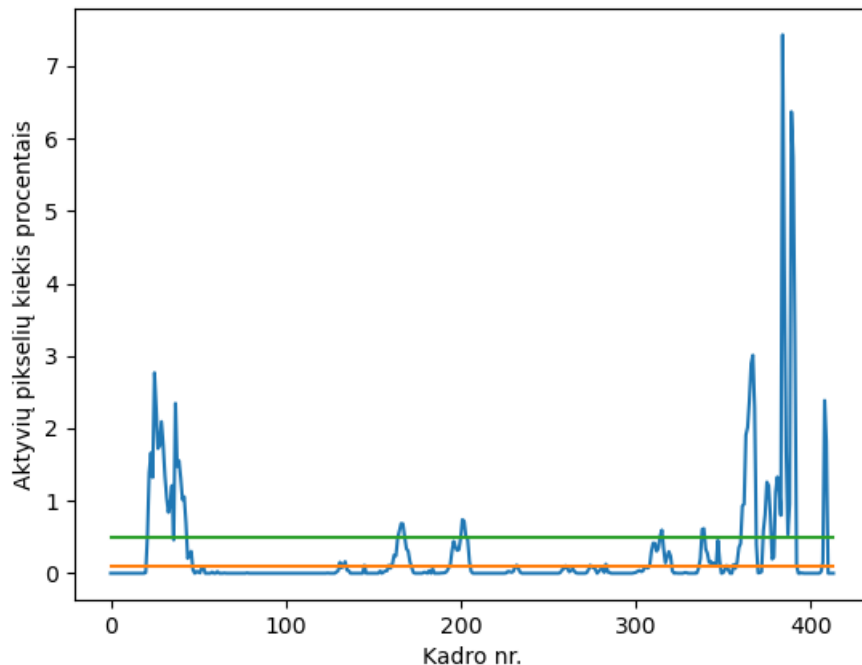
3 pav. Fono pašalinimas naudojantis Gauso skirstiniu paremtu fono segmentavimo algoritmu.

Paveikslėlyje matome, jog algoritmas atpažino žmogaus siluetą. Šiame pavyzdyje ranka priskirta fonui dėl to, jog spalva sutampa su sienos spalva, tačiau kitos kūno dalys išskiriamos iš fono.

3.2.2. Judesio atpažinimas

Žaidėjui vaikstant, kai kurios kūno dalys juda greičiau, nei kitos. Kūnas išlieka santykinai statiškas palyginus su kojomis. Žingsniavimo mechanika yra tokia, kad atliekant žingsnį, viena koja lieka vienoje vietoje, kol kita yra perstatoma iš vienos pozicijos į kitą. Tam, kad pagauti tą kojos judesį, galima paprasčiausiai iš antro kadro atimti pirmą. Padalijus kadrą į dvi dalis – viršutinę

ir apatinę – ir darant prielaidą, jog apatinėje dalyje matysis tik kojų judesiai, galima teigti, kad jei atėmus antrą kadrą iš pirmo yra skirtumas, buvo atliktas žingsnis. Koją pastačius ant žemės, tam tikrą laiką skirtumai sumažėja iki nustatytos ribos – pasinaudojus visa šia informacija galima apskaičiuoti, kiek žingsnių buvo atlikta.



4 pav. Aktyvių pikselių kiekis kiekviename kadre, indikuojantis žmogaus judėjimą.

2 pav. pateiktoje lentelėje matyti, kiek buvo aktyvių pikselių kiekvieną kadrą panaudojus fono pašalinimą bei kadrų skirtumo algoritmus. Ši informacija rodo judėjimo kiekį tam tikrame kadro regione. Žalia linija žymi minimalų judėjimo kiekį, kuris gali atsirasti dėl triukšmo ir pašalinio judėjimo. Viršijus šią liniją galima teigti, jog žaidėjas juda. Oranžinė linija žymi ribą, kada galima sakyti, jog jokio judesio nėra, t.y. žaidėjas pastatė koją ir ruošiasi atlikti kitą žingsnį.

3.3. Žmogaus kūno dalių atpažinimas

Šiuolaikinėje kompiuterinės regos srityje daug naudos galima gauti pasinaudojus neuroninių tinklų pagalba. Neuroniniai tinklai naudojami sprendžiant įvairias problemas, šiam darbui aktualiausia yra objektų atpažinimo problema. Turint tam tikrą kadrą, taisyklių pažeidimo algoritmui būtina, kad būtų atskirtos rankos, kojos ir kamuolys. Žmogaus kūno dalių klasifikavimas yra sudėtinga problema, kadangi dauguma algoritmų yra priklausomi nuo surinktų duomenų. Kompleksija tampa akivaizdi sprendžiant sporto problemas, kadangi labai dažnai žaidėjai daro įvairiausius judesius, apsirengę įvairiausiais rūbais ir pan., kas apsunkina žaidėjo kūno dalių atpažinimą [APG⁺14]. Tam reikalinga turtinga ir didelė duomenų aibė, dėl ko neuroninio tinklo apmokymo laikas išauga.

Bene visi šiuolaikiniai metodai remiasi tuo pačiu principu: apmokinamas neuroninių tinklų modelis, jiems paruošiant teigiamus ir neigiamus pavyzdžius (pvz., siekiant sukurti kojų atpažinimo algoritmą, paveikslukai, kuriuose yra koja pažymimi kaip teigiami, tie, kuriuose kojos nėra

tampa neigiamais). Mokinimo procesas gali trukti daug laiko – kartais net savaites, todėl daug paprasčiau yra surasti jau sukurta modelį ir jį pasinaudoti. Viena iš modelių buvo pasiūlytas 2018 m., pavadinimu OpenPose [CHS⁺19]. Modelio architektūra paremta konvoliuciniais neuroniniais tinklais. Modelis suskirstytas į dvi šakas. Viena jų priskiria tikimybes, kad tam tikras regionas yra tam tikra kūno dalis, kita – asociacijas tarp skirtingų kūno dalių. Atpažinimas vyksta keliais etapais, siekiant gauti kuo tikslesnius rezultatus.

Šiame darbe kuriamas vaizdo atpažinimo algoritmas kiekvieną kadrą pateikia OpenPose modeliui, o rezultate gaunamos skirtingomis spalvomis sužymėtos kūno dalys. Taisyklių pažeidimo atpažinimo algoritmui aktualios yra rankų ir pėdų sritis.



5 pav. OpenPose algoritmo pagalba atpažintos kūno dalys.

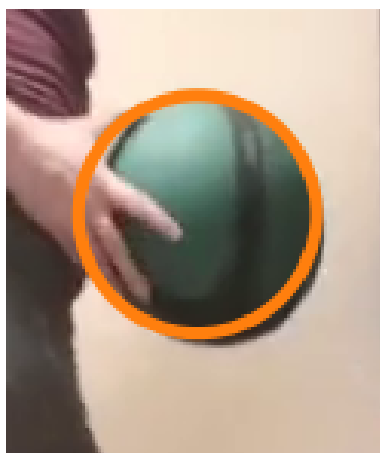
Vienas iš algoritmo trūkumų yra tai, kad be optimizavimo su kiekvienu kadru gauti rezultatą užtrunka iki 0.5 sekundės. Galimas optimizavimo būdas yra sumažinti kadro dydį prieš pateikiant jį į neuroninį tinklą, tačiau tokiu atveju rezultatų tikslumas yra atvirkščiai proporcingas kadro dimensijom.

3.4. Kamuolio atpažinimas

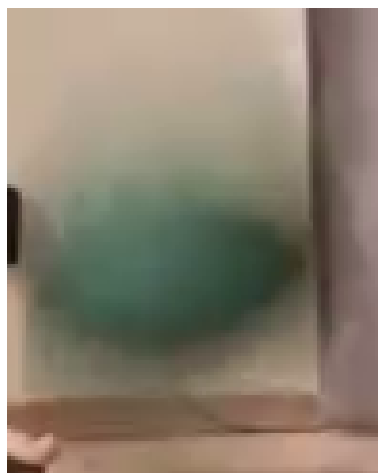
@todo fix Atpažinti kamuolį intuityviai labai paprasta. Intuitive sako, jog reikia ieškoti apvalių kontūrų. Vienas iš nagrinėtų būdų – Hough Detection. Šį algoritmą natyviai supportina OpenCV.

Hough Detection – trumpai aprašyti theory here

OpenCV turi metodą, įgyvendinantį algoritmą. Patikrinus algoritmą su vaizdo įrašu tapo akivaizdu, jog šis algoritmas tinka tik statiškam kamuoliui, o ne judančiam, kadangi jis visiškai praranda formą.



(a) Atrastas kamuolys



(b) Kamuolys deformuotas dėl judėjimo

6 pav. Fono Aaaa.

Rezultatai ir išvados

Rezultatų ir išvadų dalyje išdėstomi pagrindiniai darbo rezultatai (kažkas išanalizuota, kažkas sukurta, kažkas įdiegta), toliau pateikiamos išvados (daromi nagrinėtų problemų sprendimo metodų palyginimai, siūlomos rekomendacijos, akcentuojamos naujovės). Rezultatai ir išvados pateikiami sunumeruotų (gali būti hierarchiniai) sąrašų pavidalu. Darbo rezultatai turi atitikti darbo tikslą.

Literatūra

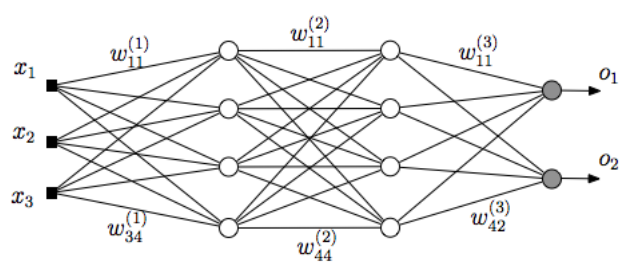
- [APG⁺14] M. Andriluka, L. Pishchulin, P. Gehler ir B. Schiele. 2d human pose estimation: new benchmark and state of the art analysis. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014-06.
- [BK08] G. Bradski ir A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc., USA, 2008.
- [CHS⁺19] Z. Cao, G. Hidalgo, T. Simon, S. Wei ir Y. Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields, 2019. arXiv: 1812.08008.
- [HSZ87] Robert M. Haralick, Stanley R. Sternberg ir Xinhua Zhuang. Image analysis using mathematical morphology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(4):532–550, 1987. DOI: 10.1109/TPAMI.1987.4767941.
- [KB02] P. KaewTraKulPong ir R. Bowden. *An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection*. Springer US, Boston, MA, 2002, p. 135–144.
- [Kuh12] D. Kuhlman. *A Python Book: Beginning Python, Advanced Python, and Python Exercises*. 2012.
- [LBC02] Na Li, Jiajun Bu ir Chun Chen. Real-time video object segmentation using hsv space. *Proceedings. International Conference on Image Processing*, tom. 2, p. II–II, 2002. DOI: 10.1109/ICIP.2002.1039893.
- [Sze21] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer International Publishing, Base, Switzerland, 2021.
- [WC15] Yi Wan ir Qiqiang Chen. Joint image dehazing and contrast enhancement using the hsv color space. *2015 Visual Communications and Image Processing (VCIP)*, p. 1–4, 2015.
- [Wel91] Emo Welzl. Smallest enclosing disks (balls and ellipsoids). *Results and New Trends in Computer Science*, p. 359–370. Springer-Verlag, 1991.

Santrumpos

Sąvokų apibrėžimai ir santrumpų sąrašas sudaromas tada, kai darbo tekste vartojami specialūs paaiškinimo reikalaujantys terminai ir rečiau sutinkamos santrumpos.

Priedas nr. 1

Neuroninio tinklo struktūra



7 pav. Paveikslėlio pavyzdys

Priedas nr. 2

Eksperimentinio palyginimo rezultatai

1 lentelė. Lentelės pavyzdys

Algoritmas	\bar{x}	σ^2
Algoritmas A	1.6335	0.5584
Algoritmas B	1.7395	0.5647