

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Bakalauro baigiamasis darbas

Patiekalo maistingumo nustatymas naudojantis neuroniniais tinklais
(food nutrition determination using neural networks)

Atliko:

Vytautas Martusevičius (parašas)

Darbo vadovas:

Lekt. Andrius Grevys (parašas)

Darbo recenzentas:

Dr. Gintautas Bareikis (parašas)

Vilnius – 2020

Santrauka

Bakalauro darbe nagrinėjamas maisto maistingumo nustatymas iš nuotraukų. Aptiriamos teorinės dalys: konvoliuciniai neuroniniai tinklai, pernešantis mokymas, Mask R-CNN tinklas, tūrio skaičiavimas iš nuotraukų. Atliekami praktiniai bandymai Mask R-CNN tinklui pritaikant aprašytas technikas bei metodus. Skaičiuojamas nuotraukose pavaizduoto maisto tūris ir maistingumas. Gauti rezultatai lyginami su kitais darbais. Sukurtas algoritmas, iš įvesties duomenų – nuotraukų, apskaičiuojanti maisto maistingumą.

Raktiniai žodžiai: Mask R-CNN, konvoliuciniai neuroniniai tinklai, maisto maistingumo nustatymas, tūrio skaičiavimas, instancijos segmentacija, pernešantis mokymas.

Summary

The bachelor's thesis deals with the determination of food nutrition from photographs. Theoretical parts are discussed: convolutional neural networks, transfer training, Mask R-CNN network, volume calculation from photographs. Practical tests are performed using the described techniques and methods for the Mask R-CNN network. The volume and nutrition of the food depicted in the photos are calculated. The obtained results are compared with other works. An algorithm is made, which from the input data - photos, calculate the nutritional value of food.

Keywords: Mask R-CNN, convolutional neural networks, food nutrition determination, volume calculation, instance segmentation, transfer training.

Turinys

Įvadas.....	6
1 Konvoliuciniai neuroniniai tinklai.....	7
2 Instancijos segmentacija.....	8
3 Mask R-CNN.....	9
3.1 Backbone (Modelio Stuburas)	9
3.2 Bruožų piramidžių tinklas (FPN).....	9
3.3 Regionų pasiūlymų tinklas (RPN)	10
3.4 Regionų pasiūlymų (ROI) klasifikatorius ir ribojančios dėžutės regresija.....	12
3.5 Segmentacijos kaukė.....	12
4 Pernešantis mokymas	13
4.1 Pernešantis mokymas mašinų mokyme	13
4.2 Pernešantis mokymas giliame mokyme.....	14
5 Mask R-CNN tinklo implementacija.....	17
5.1 Keras, Tensorflow bei Matterport.....	17
5.2 Duomenų rinkinys.....	17
5.3 Duomenų žymėjimas	18
5.4 Duomenų rinkinio paskirstymas	18
5.5 Duomenų papildymas	19
6 Tyrimo analizė.....	20
6.1 Paprasti tinklai	20
6.1.1 Mask R-CNN ResNet-50	20
6.1.2 Mask R-CNN ResNet-101	21
6.1.3 Paprastų tinklų rezultatai.....	21
6.2 Duomenų papildymas	21
6.2.1 Duomenų papildymo rezultatai	22
6.3 Tikslus derinimas.....	22

6.3.1 Tikslaus derinimo rezultatai	23
6.4 Tikslus derinimas visam tinklui	23
6.4.1 Tikslaus derinimo visam tinklui rezultatai	24
6.5 Tyrimo rezultatai.....	24
7 Tūrio apskaičiavimas	25
7.1 Elipsoidas.....	25
7.2 Kubas	26
7.3 Cilindras.....	26
7.4 Tūrio skaičiavimo rezultatai	27
8 Maisto Maistingumo nustatymas.....	29
8.1 Maistingumo nustatymo pavyzdys	30
8.2 Rezultatų palyginimas su kitais darbais	32
Išvados	34
Šaltiniai	35
Priedai	36
1 priedas. Maisto maistingumo skaičiavimas	36
2 priedas. Modelio treniravimas	36

Ivadas

Žmonės vis dažniau miršta nuo prastos mitybos sukeltų ligų. Iš anksto apdorotas, nesveikas maistas užima vis didesnę maisto dalį mūsų mityboje. Jis pigesnis bei lengviau pasiekiamas nei natūralus, sveikas maistas. Toks maistas neužpildo mūsų, todėl žmonės vis dažniau persivalgo ir nutunka. Nutukimas sukelia diabetą, įvairias kraujagyslių bei širdies ligas. Internetas pilnas nemokomos informacijos apie sveiką mitybą, maisto maistingumą bei reikalingas dienos normas, tačiau žmonės darosi vis labiau nutukę. Nors informacija nemokama ir lengvai prieinama, tačiau retas kuris iš mūsų žino kūnui reikalingu mineralų, vitaminų bei energijos kiekį. Dar rečiau skaičiuoja, kiek jau suvalgė maisto ir kokia buvo jo maistingumo vertė. Iš to išplaukia, kad žmonėms reikalingas efektyvus ir patogus būdas sekti suvartojamo maisto maistingumą.

Vis greitėjantys kompiuteriai, gerėjantys algoritmai bei išplitę išmanieji telefonai gali palengvinti suvartojamo maisto reguliavimą bei mitybos sekimą. Atsirado programų, kurioms pateikiamas valgomo maisto brūkšninis kodas. Iš brūkšninio kodo šios programos pateikia nuskenuoto maisto maistingumą. Tačiau maistas ne visada turi brūkšninį kodą. Jei valgysite kavinėje, restorane ar kokioje kitoje užėigoje, jums teks rinktis maistą iš meniu. Atsiranda poreikis nustatyti maisto maistingumą iš nuotraukų. Užtektu prieš valgant nufotografuoti maistą ir jums programa išmaniajame telefone atpažintu valgomą maisto produktą bei pateiktu jo maistingumą: cukraus kiekį, angliavandenių bei baltymų kiekį, kalorijų skaičių, vitaminų bei mineralų kiekį ir dar daugiau informacijos. Tokia programa labai supaprastintu dietos laikymąsi arba maisto maistingumo skaičiavimą. Taip pat, padėtu sergantiems tam tikromis maisto sukeltomis ligomis išvengti produktų, dėl kurių šios ligos atsirado.

Šio darbo tikslas sukurti algoritmą, gebantį iš nuotraukos nustatyti patiekalo maistingumą. Šiam tikslui pasiekti buvo nagrinėjami: gilūs konvoliuciniai neuroniniai tinklai, instancijos segmentacija, duomenų papildymas, pernešantis mokymas giliame mokyme, maisto produktų tūrio skaičiavimas iš nuotraukų. Darbo metu atlikti eksperimentai su Mask R-CNN tinklu, siekiant pagerinti algoritmo tikslumą. Gauti rezultatai lyginami su kitais algoritmais, kurie skaičiuoja maisto maistingumą iš nuotraukų. Darbo pabaigoje pateikiamos išvados ir rezultatai.

1 Konvoliuciniai neuroniniai tinklai

Bandant atpažinti ir klasifikuoti maisto produktus, reikia gebėti atpažinti sudėtingus bruožus. Šiuos bruožus atpažinti automatiškai gali konvoliuciniai neuroniniai tinklai. Šie tinklai įkvėpti gyvūnų regos žievės, kur atskiri neuronai reaguoja į persidengiančius matymo lauko regionus. Būtent šis panašumas paverčia konvoliucinius neuroninius tinklus ypatingai tinkamus kompiuterių vizijai (sritis, kurioje bandoma automatizuoti užduotis, kurias geba atlikti žmogaus rega), kadangi gyvūno regos bei kompiuterių vizijos užduotis ta pati – gauti supratimą iš nuotraukų.

Kai nuotrauka yra pateikiama konvoliuciniam neuroniniam tinklui, atliekama eilė operacijų duomenims einant per tinklą. Kiekvieno sluoksnio parametrai yra koreguojami kiekvienos iteracijos metu – taip mokosi tinklas. Konvoliucinis neuroninis tinklas sudarytas iš tam tikrų sluoksnių:

- Konvoliucijos sluoksnis – sluoksnis sudarytas iš mokymo filtrų. Šie filtrai mokomi tinklo taip, kad reaguotų į tam tikrus bruožus nuotraukoje. 1 paveikslėlyje pavaizduotas išmoktų filtrų pavyzdys.
- Susiejimo sluoksnis – mažina erdvinį duotosios nuotraukos dydį.

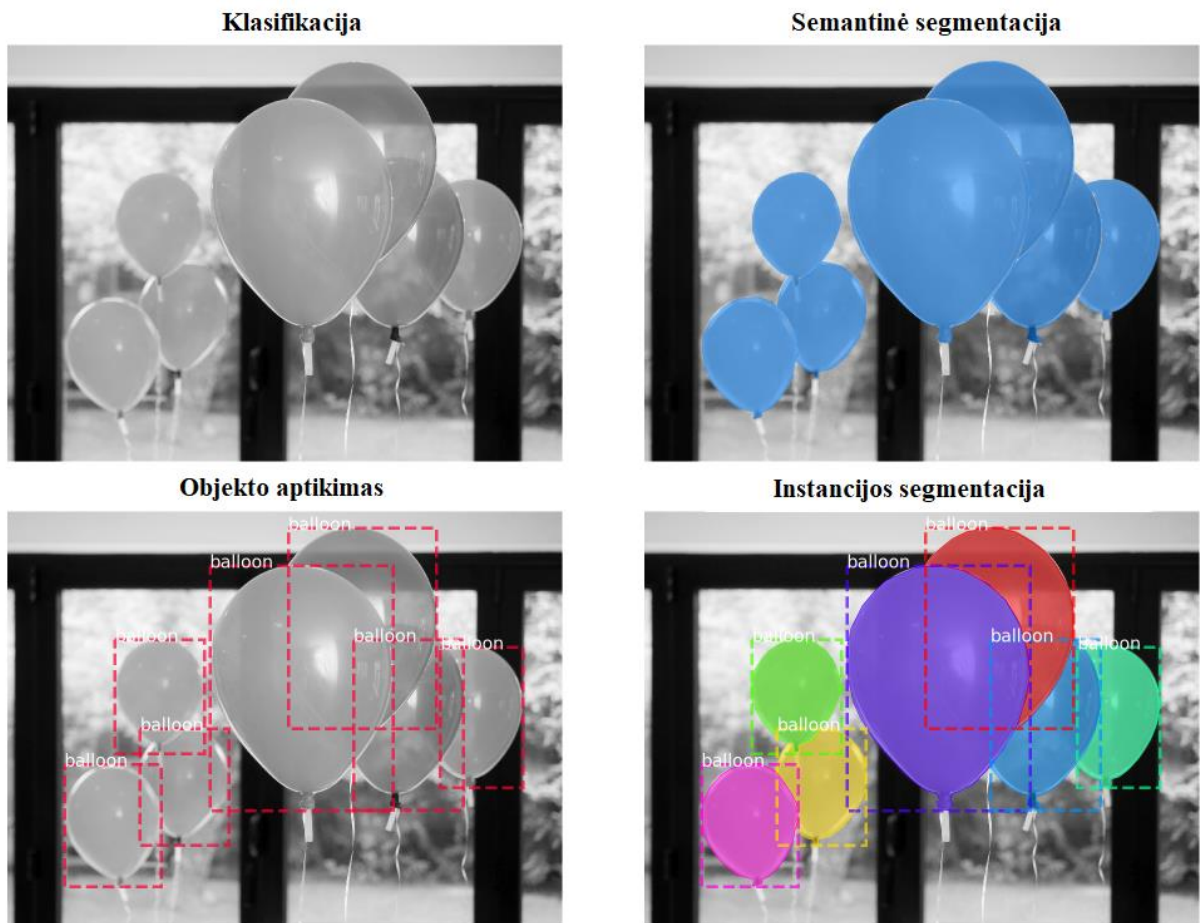


pav. 1: pavaizduoti išmokti bruožai, kuriuos išmoko pirmo konvoliucinių sluoksnio filtrai. Bruožai nesudėtingi, pavyzdžiui kampų pasukimai. [NU17]

Gerinti konvoliucinio tinklo tikslumą galima didinant tinklo dydį: vis pridedant tinklui daugiau sluoksnių, sluoksniuose daugiau neuronų. Vis didėjant tinklam atsiranda nauja sąvoka – gilūs neuroniniai tinklai. Norint apmokyti šiuos tinklus reikia turėti labai daug resursų: kompiuterio galios, laiko, didelio duomenų rinkinio.

2 Instancijos segmentacija

Kompiuterių vizija sprendžia ne tik objekto klasifikavimo problemą, bet ir jo aptikimą erdvėje. Objektų aptikimo algoritmai ir klasifikavimo algoritmai skiriasi tuo, kad aptikimo algoritmuose mes stengiamės nubrėžti apribojančią objektą dėžę aplink dominantį objektą, kad jį rastume atvaizde. Instancijos segmentacija yra objekto kontūrų aptikimas pikselių lygyje. Tai yra viena sunkiausia kompiuterių vizijos sprendžiama problema. 2 paveikslėlyje pavaizduotas šios problemos sudėtingumas lyginant su kitomis kompiuterių vizijos problemomis:



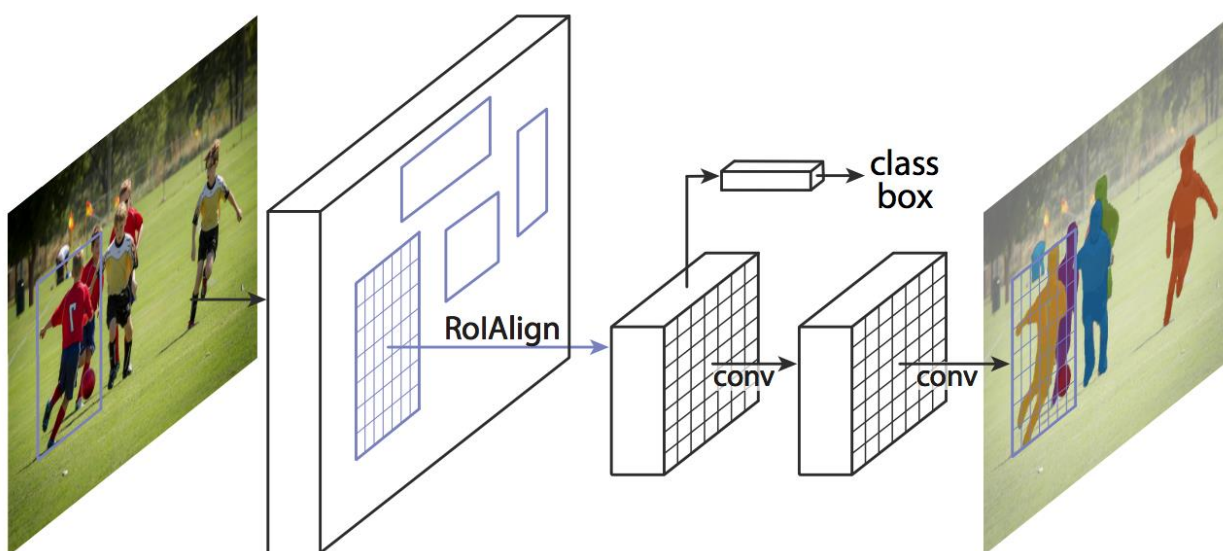
pav. 2 Kompiuterių vizijos sprendžiamos užduotys

- Klasifikacija: Ar nuotraukoje yra balionas?
- Semantinė segmentacija: Čia yra visi pikseliai, kuriuos sudaro balionas.
- Objekto aptikimas: Nuotraukoje yra 7 balionai ir jų padėtys nuotraukoje yra apibrėžtos raudonu kontūru.
- Instancijos segmentacija: Nuotraukoje yra 7 balionai padėtyse apibrėžtose spalvotais kontūrais. Ir kiekvienas balionas turi skaičių, iš kiek pikselių jis sudarytas. Šis pikselių skaičius vadinamas kauke (angliškai mask).

Maisto maistingumo nustatymo užduotį sprendžiau instancijos segmentacijos metodu, naudodamas Mask R-CNN modelį.

3 Mask R-CNN

Mask R-CNN (regioninis konvoliucinis neuronų tinklas) yra dviejų pakopų sistema: pirmoji pakopa nuskaityto vaizdą ir generuoja pasiūlymus (sritis, kuriose gali būti objektas). Antrasis etapas klasifikuoja pasiūlymus ir sukuria ribojančias dėžutes ir kaukes. Mask R-CNN yra tų pačių autorių sukurto Faster R-CNN tinklo pratęsimas. Faster R-CNN yra populiarus modelis objektų aptikimui, o Mask R-CNN tinklas jį pagerina ir papildo su instancijos segmentacija. 3 paveikslėlyje pavaizduota šiuo modelio architektūra:



pav. 3 Mask R-CNN modelio architektūra [MaskRCNN]

Mask R-CNN sudarytas iš šių modulių:

3.1 Backbone (Modelio Stuburas)

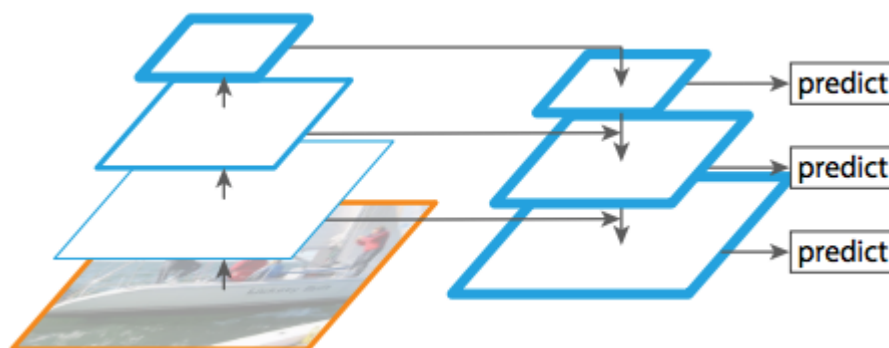
Šis modulis yra standartinis konvoliucinis neuroninis tinklas (dažniausiai naudojamas yra ResNet50 arba ResNet101 tinklas) kuris veikia kaip bruožų iš traukiklis (toliau apie bruožų ištraukimus skyriuje pernešantis mokymas). Ankstyvieji sluoksniai aptinka žemo lygio bruožus (kraštus ar kampus), o vėlesni sluoksniai iš eilės nustato aukštesnio lygio bruožus (automobilis, asmuo, bananas, dviratis).

3.2 Bruožų piramidžių tinklas (FPN)

Anksčiau aprašytas modelio stuburas (angliškai backbone) puikiai veikia, tačiau jį galima patobulinti. Šio Mask R-CNN modelio autoriai pristatė bruožų piramidžių tinklą (angliškai feature

pyramid network arba trumpiau FPN) kaip plėtinį, galintį geriau atvaizduoti objektus keliuose skalėse.

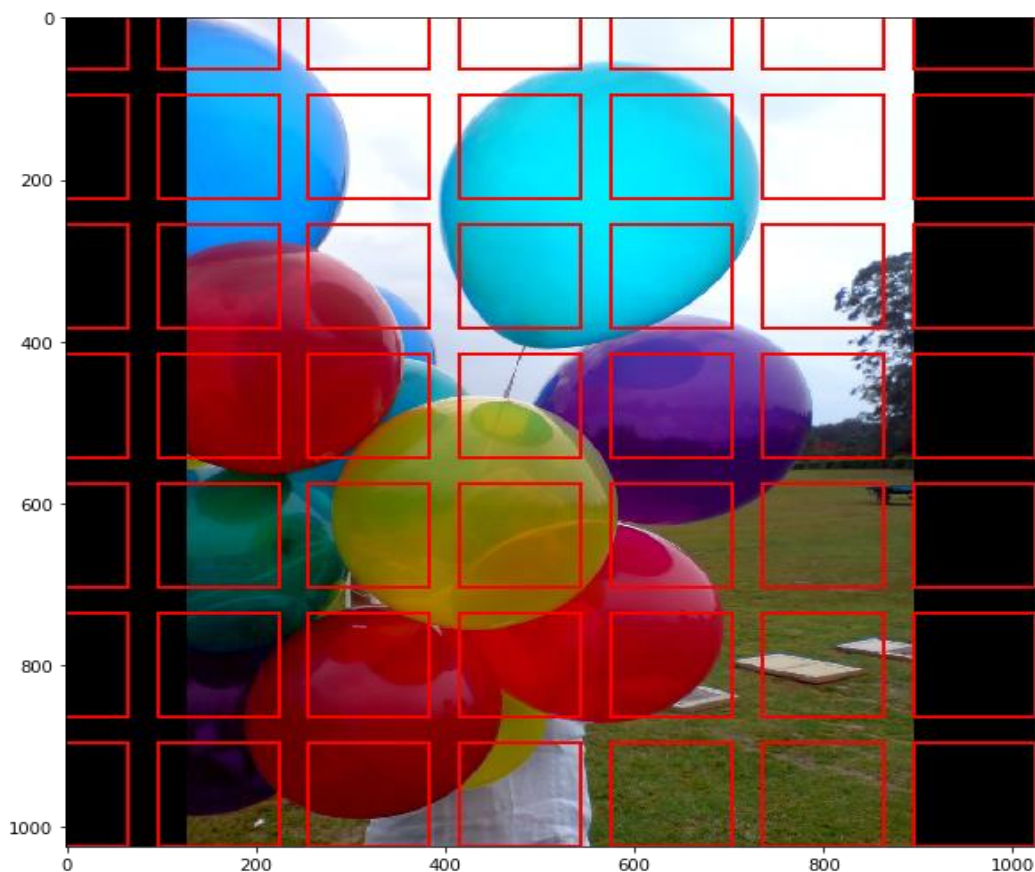
Bruožų piramidžių tinklas pagerina standartinį bruožų ištraukimą, pridėdam antrą piramidę, kuri paima aukšto lygio bruožus iš pirmosios piramidės ir perduoda jas žemyn į apatinius sluoksnius. Tai leidžia kiekvieno lygio bruožams pasiekti tiek žemesnio, tiek aukštesnio lygio bruožus. 4 paveikslėlyje pavaizduotas bruožų piramidžių tinklas:



pav. 4 bruožų piramidžių tinklas [FPN]

3.3 Regionų pasiūlymų tinklas (RPN)

Regionų pasiūlymų tinklas yra lengvas neuroninis tinklas, skenuojantis vaizdą slankiojančiu langu ir randantis sritis, kuriose yra objektai. Regionai, kuriuos nuskaito regionų



pav. 5 supaprastinta iliustracija su 49 inkarų dėžutėmis

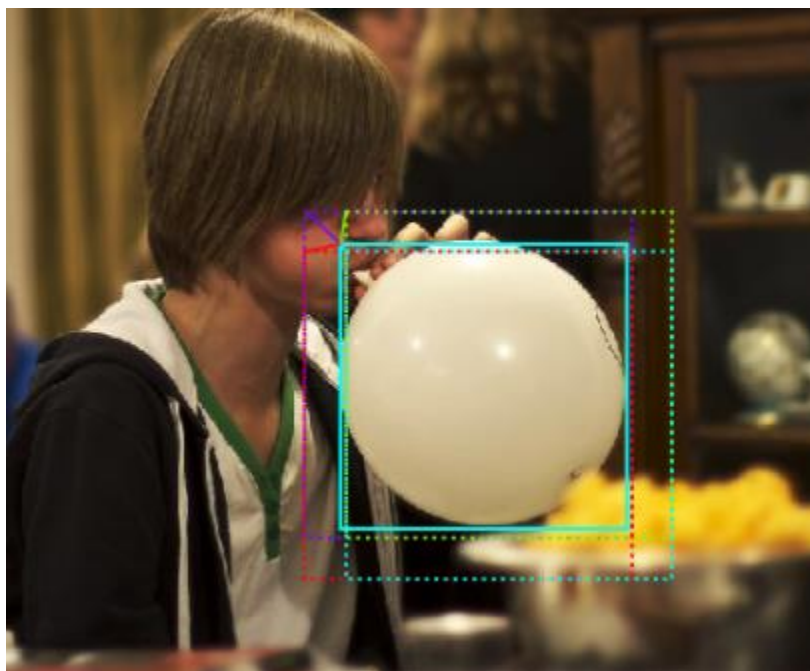
pasiūlymu tinklas, vadinami inkarais (angliškai anchors). Šie inkarai yra dėžutės, paskirstytos per visą nuotraukos sritį kaip pavaizduota 5 paveikslėlyje.

Tačiau 5 paveikslėlyje pateiktas inkarų pavyzdys yra supaprastintas. Dažniausiai yra apie 200 tūkstančių įvairaus dydžio ir kraštinių santykio inkarų. Taip pat jie persidengia, kad apimtu kuo didesnę nuotraukos sritį.

Nors 200 tūkstančių inkarų atrodo daug, tačiau jie nuskaitomi gana greitai. Stumdomas langas yra valdomas atsižvelgiant į regionų pasiūlymų tinklo konvoliucinį pobūdį, kuris leidžia jam nuskaityti visus regionus lygiagrečiai (naudojantis GPU). Be to, regionų pasiūlymų tinklas neskenuoja vaizdo tiesiogiai. Vietoj to, jis nuskaito modelio stuburo sudaryta bruožų žemėlapi. Tai leidžia regionų pasiūlymų tinklui efektyviai pakartotinai panaudoti ištrauktus bruožus ir išvengti pasikartojančių skaičiavimų. Atlikus šią tinklo optimizaciją, tinklas gali veikti 10 ms greičiu, kaip nurodyta Faster R-CNN popieriuje [FASTERRCNN], kuris ir pristatė šį regionų pasiūlymų tinklą. Tačiau Mask R-CNN naudoja didesnes nuotraukas ir daugiau inkarų, todėl tai gali užtrukti kiek ilgiau.

Regionų pasiūlymų tinklas sukuria dvi išvestis kiekvienam inkarui:

1. Inkaro klasė: priekinio plano (angliškai foreground) arba foninė (angliškai background). Priekinio plano klasė reiškia, kad toje dėžutėje gali būti objektas.
2. Ribojančios dėžutės patikslinimas: priekinio plano inkaras (dar vadinamas teigiamu inkaru) gali būti ne visiškai centruotas ties objektu. Todėl regionų pasiūlymų tinklas



pav. 6 pavaizduotos 3 inkaro dėžutės (taškuotos) ir jų pokytis, pritaikytas pavaizduoti objektą tiksliai (vientisa linija). Keli inkarai gali rodyti tą patį objektą.

įvertina pločio bei aukščio pokyčius ir patobulina inkaro dėžutę, kad ji tiksliau nurodytu objekto poziciją. Tai pavaizduota 6 paveikslėlyje.

Naudodamiesi regionų pasiūlymų tinklo spėjimais, pasirenkame geriausius inkarus, kuriuose greičiausiai yra objektų. Tuomet patiksliname jų vietą bei dydį. Jei keli inkarai persidengia per daug, pasilieiname tą, kuriame yra didžiausias priekinio plano rezultatas, o likusius išmetame. Lieka tik galutinius pasiūlymai (dominantys regionai), kuriuos perduodame kitam etapui.

3.4 Regionų pasiūlymų (ROI) klasifikatorius ir ribojančios dėžutės regresija

Šis etapas vykdomas regionų pasiūlymuose (ROI's – angliškai regions of interest), kuriuos pasiūlė regionų pasiūlymų tinklas. Šis etapas generuoja dvi išvestis kiekvienam regiono pasiūlymui:

1. Klasė: objekto klasė regiono pasiūlyme. Skirtingai nuo regionų pasiūlymo tinklo, kuris turi dvi klases – fono ir priekinio plano, šis tinklas yra gilesnis ir turi galimybę regionus suskirstyti į tam tikras klases (asmuo, automobilis, kėdė, bananas...). Taip pat gali sukurti foninę klasę, dėl kurios pasiūlytas regionas gali būti atmestas.
2. Apribojimų dėžutės patikslinimas: labai panašu į tai, kaip daroma regionų pasiūlymu tinkle. Jos tikslas yra dar labiau patikslinti ribojančios dėžutės vietą ir dydį, taip nurodant tikslią objekto vietą nuotraukoje.

Tačiau čia susiduriame su problema - klasifikatorius netvarko kintamo įvesties dydžio. Įprastai į klasifikatorių ateina fiksuotas įvesties dydis. Tačiau, dėl ribojančių dėžučių patobulinimo regionų pasiūlymo tinklo žingsnyje, regionų pasiūlymų dėžutės gali būti skirtingo dydžio. Čia ir naudojamas „ROI pooling“. Jis apkarpo bruožų žemėlapi ir pakeičia jo dydį iki fiksuoto dydžio. Iš esmės tai panašu į vaizdo apkarpyimą ir jo dydžio pakeitimą.

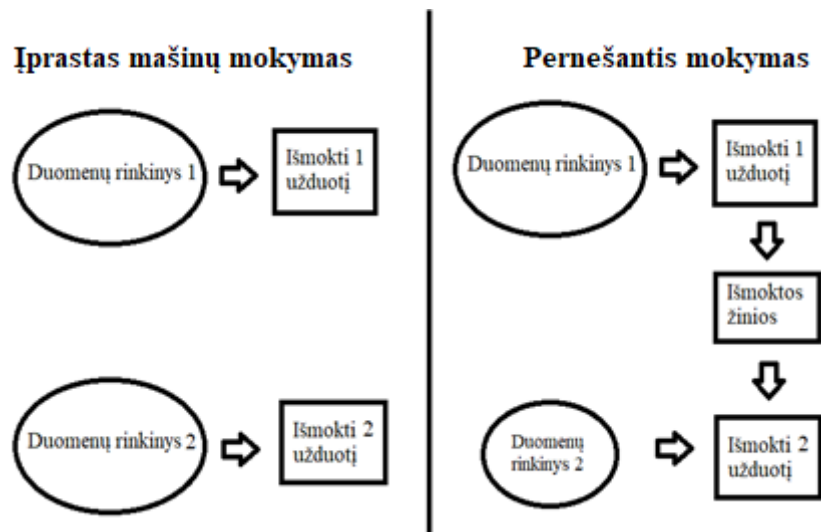
3.5 Segmentacijos kaukė

Jei sustotume ties 4.4 skyriaus pabaiga, turėtumėme Faster R-CNN modelį objektų aptikimui. Kaukių tinklas yra priedas, kurį pristatė Mask R-CNN popierius [MASKRCNN]. Kaukės šaka yra konvoliucinis tinklas, kuris paima teigiamus regionus, kuriuos pasirinko regionų pasiūlymų klasifikatorius, ir sukuria jiems kaukes.

4 Pernešantis mokymas

4.1 Pernešantis mokymas mašinų mokyme

Dažniausiai konvoliuciniai neuroninių tinklų modeliai apmokomi dirbti tiktais tam tikroje srityje. Nors tam tikroje srityje tam tikras modelis gali būti pats geriausias, tačiau jis bus toks tikslus dirbant tik su tam tikrais duomenų rinkiniais ir veiks žymiai prasčiau bandant jį pritaikyti panašiai, tačiau naujai užduočiai. Šioje srityje ir ateina pernešantis mokymas, kuris leidžia labai gerus tinklus, išmokytus ant labai didelio duomenų rinkinio, pritaikyti panašiai užduočiai. Šį metodą galima palyginti su žmogišku mokymusi. Žmonės visko nesimoko nuo pradžių, pavyzdžiui, moksleivis, išmokęs matematikos, per naują jos nesimoko mokydamasis fizikos. Taip pat muzikantas, išmokęs groti klasikine muziką pianinu, mokydamasis groti džiazo stiliaus muziką, iš naujo nesimokys groti pianinu, o tik panaudos savo jau turimas žinias ir pritaikys jas kitam stiliui.



pav. 7 Įprastas mašinų mokymas prieš pernešantį mokymą

Paveikslėlyje 7 pavaizduotas įprastas mašinų mokymasis ir pernešantis mokymas. Įprastas mokymas yra izoliuotas ir skirtas tik specifinėms užduotims su tam tikrais duomenų rinkiniais, jokios išmoktos žinios nėra išsaugomos, todėl kitą kartą atliekant panašią užduotį modelį reikia mokyti vėl nuo pradžių su kitokiu duomenų rinkiniu. Tačiau naudojantis pernešančio mokymo metodika, galima išgauti žinias (pavyzdžiui išmoktus bruožus bei svorius) iš prieš tai apmokytu modelių ir įveikti tokias kliūtis kaip didelio duomenų rinkinio nebuvimas. Pavyzdžiui, modelis T1, skirtas atpažinti lengvuosius automobilius. Šis modelis gali veikti labai gerai ir būti labai tikslus, tačiau su juo bandant atpažinti sunkvežimius, jis veiks žymiai blogiau nei veikė anksčiau. Tačiau pritaikius pernešantį mokymą ir nedidelį sunkvežimių duomenų rinkinį, galime sukurti naują modelį T2, kuris bus išmokytas sunkvežimius atpažinti ne prasčiau nei modelis T1 atpažino

lengvuosius automobilius. Naudojant pernešamą mokymą, reikia nagrinėti tris svarbius klausimus [ASTL]:

- Ką perkelti? Šis klausimas yra be galo svarbus. Jo metu sprendžiame, kuri žinių dalis gali būti perkelta iš anksčiau apmokyto į naują modelį taip, kad pagerintume paskesnio tikslumą. Bandant atsakyti į šį klausimą, svarbu atsižvelgti į bendrus bruožus tarp modelių ir atskirti tik pirmajam modeliui priklausančius bruožus.
- Kada perkelti? Gali būti atveju, kad perkeldami išmoktas žinias, tik pabloginame modelio tikslumą. Toks atvejis dar vadinamas negatyviu perkėlimu (angliškai „negative transfer“). Mūsų tikslas pagerinti modelio tikslumą, todėl turime būti atsargūs kada perkėlinėti žinias, kada ne.
- Kaip perkelti? Kai jau nustatėme ką ir kada perkelsime iš jau išmoktų žinių, tuomet reikia pradėti nagrinėti būdus, kaip tas žinias pritaikyti naujam modeliui. Gali tekti atlikti pakeitimus algoritmams.

Apibendrinant, pernešantis mokymas leidžia naudoti prieš tai apmokytų modelių žinias ir pritaikyti naujiems, panašias užduotis sprendžiantiems modeliams. Jei modeliui T1 turime didelį duomenų kiekį, tai spręsti panašiai užduočiai sukuriame modelį T2 ir modelio T1 išmoktas žinias pritaikome užduočiai, kurią sprendžia modelis T2.

Yra skirtingų pernešančio mokymo technikų bei strategijų. Šios strategijos naudojamos atsižvelgiant į tai, su kokiais duomenimis dirbama, kokia yra užduotis ar kiek turime duomenų. Toliau kalbėsiu apie pernešama mokymą giliuose neuroniniuose tinkluose. Giliame mokyme tinka induktyvus pernešamas mokymas – abiejų modelių duomenų rinkiniai panašūs, tačiau skiriasi užduotys. Naudojamasi pirminio modelio žiniomis, bandant pagerinti antrąjį modelį.

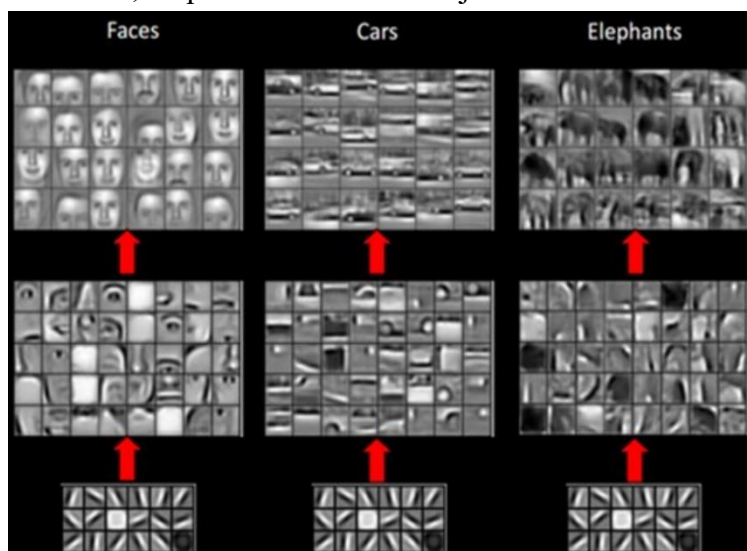
4.2 Pernešantis mokymas giliame mokyme

Pastaraisiais metais gilius mokymas padarė didelį progresą, tačiau treniruoti tokius modelius reikia daug resursų: mokymo laiko bei didelio duomenų rinkinio. Yra nemažai gilaus modelio tinklų, kurie geba atpažinti objektus neprasčiau nei žmonės. Pats svarbiausias pernešančio mokymo reikalavimas – modelis, veikiantis gerai su mūsų ar panašia į mūsų užduotimi. Laimei, komanda atradusi naują tinklo architektūrą ar sugebėjusi pagerinti modelio tikslumą, šiuo modeliu pasidalina su pasauliu. Todėl šie geriausi modeliai (angliškai „state-of-the-art“) yra pasiekiami ir mums, juos galima pasiekti netgi apmokytus su tam tikru duomenų rinkiniu. Tokio rinkinio geras pavyzdys yra ImageNet duomenų rinkinys, kuris sudarytas iš 1000 kategorijų, 1200000 nuotraukų

bei 100000 testavimo skirtų nuotraukų. Šiuos iš anksto apmokytus modelius naudoja dviem būdais:

- „Feature extraction“ - bruožu ištraukimas.
- „Fine tuning“ – tikslus derinimas.

Gilaus mokymo modeliai yra sluoksniuotos architektūros. Šiuose sluoksniuose modelis išmoksta skirtingus bruožus skirtinguose sluoksniuose. Galiausiai šie sluoksniai sujungiami į paskutinį sluoksnį, kuriame gaunami išeities duomenys (atsakymas). Ši sluoksniuota architektūra leidžia naudotis iš anksto apmokytais modeliais, tokiais kaip ResNet-50. Modelio išmoktos žinios užšaldomos sluoksniuose ir jie gaudami naujus duomenis nesimoko iš jų. Toks modelio panaudojimas vadinamas bruožu ištraukimu. Naudojamasi šio modelio konvoliucijos sluoksniais, kurie jau yra išmokę tam tikrus bruožus. Kaip pavaizduota 1 paveikslėlyje, pirmieji konvoliuciniai sluoksniai išmoksta tik tam tikrus paprastus bruožus, kaip briaunas ar kraštines. Modeliui gilėjant, sudėtingėja ir bruožai. Paskutiniai sluoksniai bando spręsti užduotį - atpažinti, klasifikuoti objektą. 8 paveikslėlyje pavaizduota, kaip žemesni konvoliucijos sluoksniai išmoksta atpažinti kraštus bei



pav. 8 skirtingi bruožai, kuriuos išmoksta konvoliucinis tinklas skirtinguose lygiuose. [MLFE]

figūras. Vidutinio lygio išmoksta atpažinti nedidelius objektus. Aukščiausio lygio sluoksniai sugeba atpažinti visą klasifikuojamą objektą arba sprendžia jiems skirtą užduotį.

Galime paimti iš anksto apmokytą modelį ir jį naudoti ne tik kaip bruožų ištraukiklį, tačiau leisti jo sluoksniams mokytis. Dažniausiai užšaldomi visi modelio sluoksniai, nesuteikiant jiems galimybės mokytis. Po to, pasirenkama keliems sluoksniams leidžiama mokytis. Toks modelio paskutiniųjų sluoksnių atšaldymas (leidimas jiems mokytis) vadinamas tikslus derinimas. Šis metodas reikalauja daugiau resursų bei pastangų, kadangi modelis turi vėl mokytis. Dažnai taikant tikslų derinimą, reikia sumažinti modelio mokymosi sparta (angliškai „learning rate“) tam, kad

modelis nepersimokytu (angliškai „overfit“). Jei modelis persimoko, tuomet jis išmoksta labai gerai atpažinti turimo duomenų rinkinio nuotraukas, tačiau pateikus naujus duomenis (validacijos arba testavimo), jo spėjimai būna prasti.

Baigiamajam bakalauro darbui pasirinkau pernešantį mokymą, kadangi giliam mokymui reikalinga labai daug duomenų. Tačiau gauti šiuos duomenis bei su jais dirbti reikia labai daug laiko bei resursų.

5 Mask R-CNN tinklo implementacija

5.1 Keras, Tensorflow bei Matterport

Mask R-CNN modelis realizuotas naudojantis Python 3.6 versiją. Panaudotos dvi Python bibliotekos bruožų mokymuisi: „Keras“ ir „Tensorflow“. Naudojau „Matterport“ Github saugyklą [Matterport] sukurti unikalų modelį skirta maisto segmentacijai. Šią saugyklą sudaro:

1. Atviras Mask R-CNN kodas;
2. Mokymo kodas MS COCO duomenų rinkiniui;
3. Iš anksto apmokyti MS COCO svoriai;

Tensorflow yra nemokama ir atvirojo kodo programinės įrangos biblioteka, skirta duomenų srautams ir diferencijuotam programavimui įvairiose užduotyse. Tai yra simbolinė matematikos biblioteka, taip pat naudojama mašininio mokymosi programoms, tokioms kaip neuroniniai tinklai.

Keras – aukšto lygio neuroninių tinklų API, parašyta Python programavimo kalba ir galinti veikti su Tensorflow.

5.2 Duomenų rinkinys

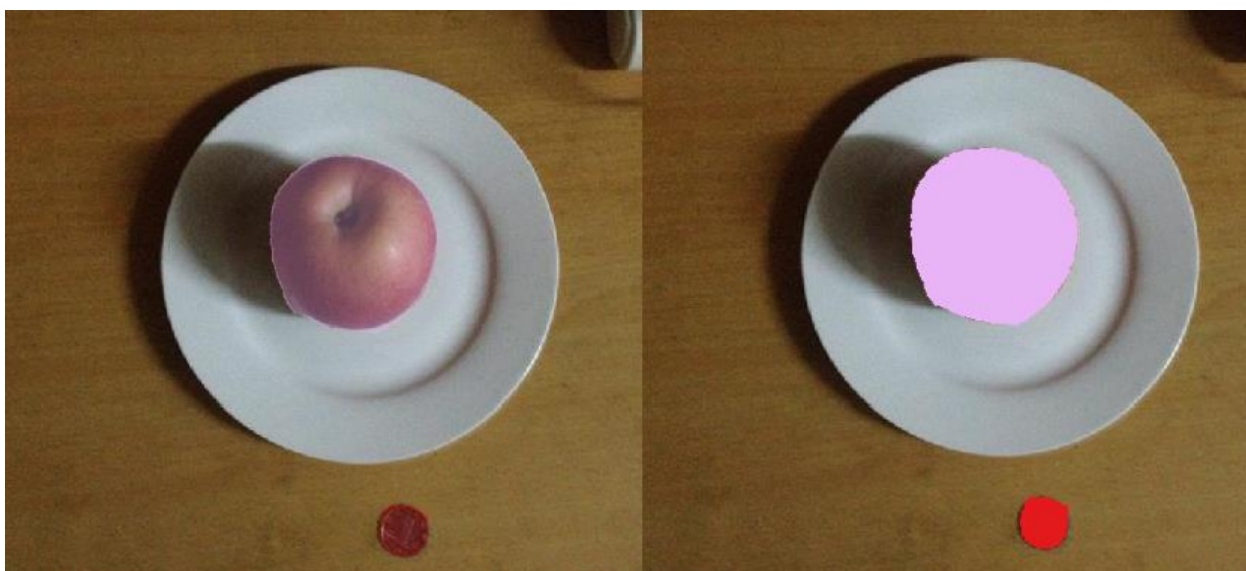
Naudojau „ECUSTFD“ [ECUSTFD] duomenų rinkinį. Jis yra nemokamas bei viešas. Šis duomenų rinkinys sudarytas iš 19 maisto rūšių ir beveik 3000 maisto nuotraukų. Visos nuotraukos darytos su išmaniuoju telefonu. Kiekvienai maisto porcijai yra padaryta kelios nuotraukos: vaizdas iš viršaus, ir vaizdas iš šono. Kiekvienoje nuotraukoje yra moneta, kaip kalibravimo objektas ir ne daugiau kaip 2 maisto produktai. Jei tame pačiame paveikslėlyje yra du maisto produktai, vieno maisto rūšis skiriasi nuo kito. 9 paveikslėlyje pavaizduotos kelios duomenų rinkinio nuotraukos:



pav. 8 banano nuotrauka iš ECUSTF duomenų rinkinio. Paveikslėlyje matome dvi banano nuotraukas, viena iš viršaus, kita iš šono. Taip pat, kiekvienoje nuotraukoje yra po monetą.

5.3 Duomenų žymėjimas

Nors ir turėjau duomenų rinkinį, tačiau mokant tinklą instancijos segmentavimui, reikia sužymėtų duomenų. Žymėti duomenys reiškia, kad kiekvienai nuotraukai kuri patenka į tinklą mokymuisi ar validacijai turi būti nurodyta joje esančių objektų jos kaukė bei klasė. Tai atlikau su „Hasty GmbH“, ši įranga laisvai prieinama internete. Į ją įkėliau visas nuotraukas kurių man reikėjo ir pradėjau žymėti viena po kitos. Kiekvienai nuotraukai reikėjo nurodyti joje esančius objektus bei apibraukti to objekto pikselių vietą – kaukę. Vienas iš pažymėtos nuotraukos pavyzdžių pateiktas 10 paveikslėlyje:



pav. 9 kairėje nuotrauka su obuoliu ir moneta. Žymint duomenis reikėjo nurodyti kaukės vieta - nuspalyta dešinės nuotraukos dalis. Taip pat reikėjo nurodyti skirtingas maisto klases: raudona - moneta, rožinė - obuolys.

5.4 Duomenų rinkinio paskirstymas

Galutinis duomenų rinkinys buvo sudarytas iš 6 klasių: obuolys, bananas, duonos riekė, virtas kiaušinis, pomidoras, moneta. Kiekvienoje nuotraukoje buvo po maisto objektą ir monetą. Lentelėje 1 pavaizduota, kiek kiekvienai klasei treniravimui bei validacijai buvo skirta sužymėtų nuotraukų.

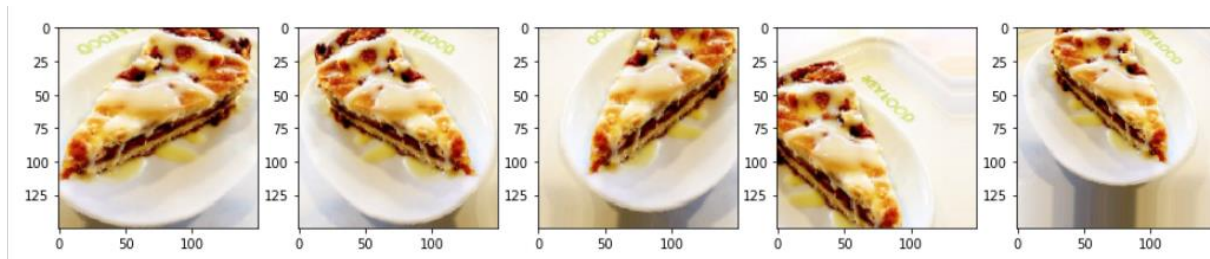
Klasė	Treniravimui	Validacijai
Obuolys	50	10
Bananas	53	10
Duonos riekė	31	10
Virtas kiaušinis	38	10
pomidoras	40	10
moneta	212	50

Lentelė 1 treniravimo bei validacijos duomenys kiekvienai klasei

5.5 Duomenų papildymas

Kai duomenų rinkinyje yra mažai treniravimo duomenų kiekvienai klasei, tuomet besimokydamas modelis dirba su vis tomis pačiomis nuotraukomis. Galiausiai jis išmoksta dirbti su treniravimo nuotraukomis, o naujų, pavyzdžiui, validacijos ar testavimo, nuotraukų nesugeba tiksliai atpažinti. Ši mažo treniravimo duomenų rinkinio problema sprendžiama duomenų papildymu (angliškai „data augmentation“). Duomenų papildymo metu mes paimame treniravimo duomenis ir jiems pritaikome kokias nors transformacijas: sukimus, mastelio keitimus, horizontalius bei vertikalius nuotraukų apvertimus ir dar daugiau įvairių nuotraukų transformacijų. Atlikus šias transformacijas, iš vienos treniravimo nuotraukos galime gauti kelias ar keliolika panašių nuotraukų į ją, tačiau vienodų niekada negausime.

Paveikslėlyje 11 pavaizduotas duomenų papildymas pritaikytas obuolių pyrago nuotraukai. Šiam pyrago gabalėliui buvo pritaikytas mastelio keitimas, posūkis, vertikalus posūkis bei aukščio ir ploto pokyčiai. Kiekvienai nuotraukai pritaikę duomenų papildymą, vietoj jos naudojame transformuotas nuotraukas.



pav 11 Obuolių pyragui pritaikytas duomenų papildymas.

6 Tyrimo analizė

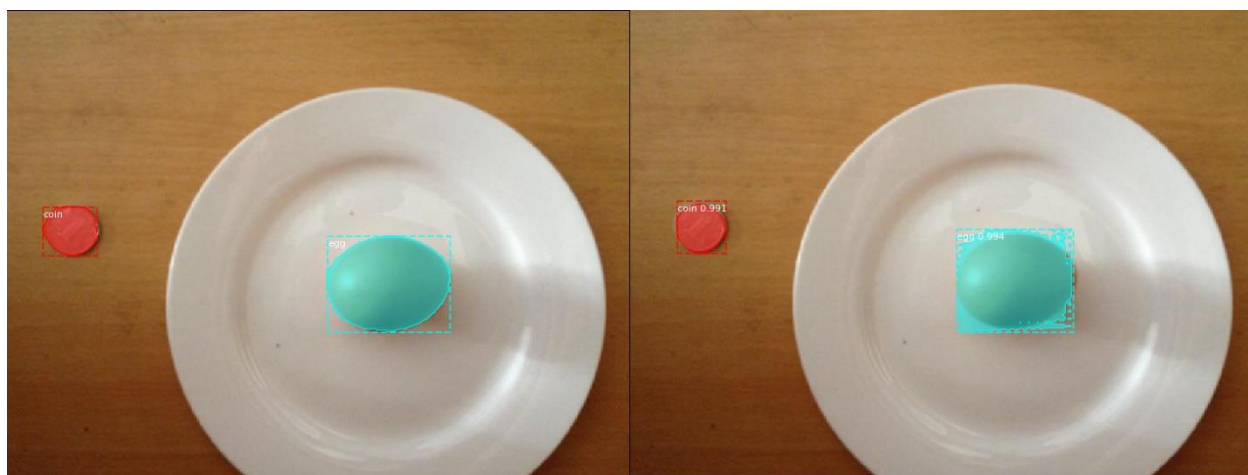
Šioje dalyje aprašyta tyrimo analizė. Joje išbandžiau anksčiau aprašytas technikas tinklo treniravimui. Visus tinklus treniravau su tuo pačiu savo sudarytu ir sužymėtu duomenų rinkiniu, kurio duomenys buvo pasiskirstę kaip nurodyta 5.4 skyriuje, 1 Lentelėje. Visų tinklų užduotis ta pati – atpažinti objektus, juos klasifikuoti iš galimų 6 klasių (obuolys, pomidoras, kiaušinis, duonos riekė, bananas, moneta) ir nurodyti jų buvimo vieta nuotraukoje taip pat nurodant kaukę. Kiekvienas tinklas įvesties duomenims gavo nuotrauką, kurios buvo (512, 512, 3) dydžio. 512 reiškia nuotraukos dydį pikseliais į aukštį ir plotį, 3 reiškia, kad nuotraukos buvo spalvotos, 3 kanalų – raudona, žalia ir mėlyna. Kiekvieno eksperimento pabaigoje pateikiamas loss – skaičius, nurodantis, koks blogas buvo modelio spėjimas. Jei modelio spėjimas buvo tobulas – loss yra 0, kitu atveju loss yra didesnis. Loss leidžia įvertinti tinklo padarytas klaidas prognozuojant rezultatus. Taip pat pateikiamas mAp - vidutinis tikslumas (angliškai mean average precision), kuo mAp arčiau 1, tuo modelis tikslesnis. vidutinis tikslumas įvertina modelio gerumą visam duomenų rinkiniui.

6.1 Paprasti tinklai

Šis eksperimentas buvo atliktas, įvertinti paprasto tinklo mokymąsi be papildomų technikų, taip pat nustatyti, kurį tinklo stuburą (backbone) rinktis – ResNet-50 ar ResNet-101.

6.1.1 Mask R-CNN ResNet-50

Šis tinklas yra Mask R-CNN tinklas, treniruotas su 5.4 skyriaus 1 lentelės duomenų rinkiniu. Treniruojama buvo 15 epochų, treniruojami visi tinklo sluoksniai. Tinklo stuburas (backbone) buvo ResNet-101 modelis. Šis modelis sudarytas iš 101 sluoksnių. Modelis per 15 epochų pasiekė 0.9045 loss ir mAp – 0.75. Modelio spėjimai pavaizduoti 12 paveikslėlyje.



pav. 12 sužymėta nuotrauka kairėje, modelio spėjimas dešinėje

6.1.2 Mask R-CNN ResNet-101

Šis tinklas yra Mask R-CNN tinklas, treniruotas su treniruotas su 5.4 skyriaus 1 lentelės duomenų rinkinių. Treniruojama buvo 15 epochų, treniruojami visi tinklo sluoksniai. Tinklo stuburas (backbone) buvo ResNet-101 modelis. Šis modelis sudarytas iš 101 sluoksnių. Modelis per 15 epochų pasiekė 0.8928 loss ir mAp – 0.75. Iš paveikslėlio 13 matome, kad modelio klasifikavimai buvo teisingi, tačiau kaukės ir dėžutė nurodyta neteisingai.



pav. 103 sužymėta nuotrauka kairėje, modelio spėjimas dešinėje

6.1.3 Paprastų tinklų rezultatai

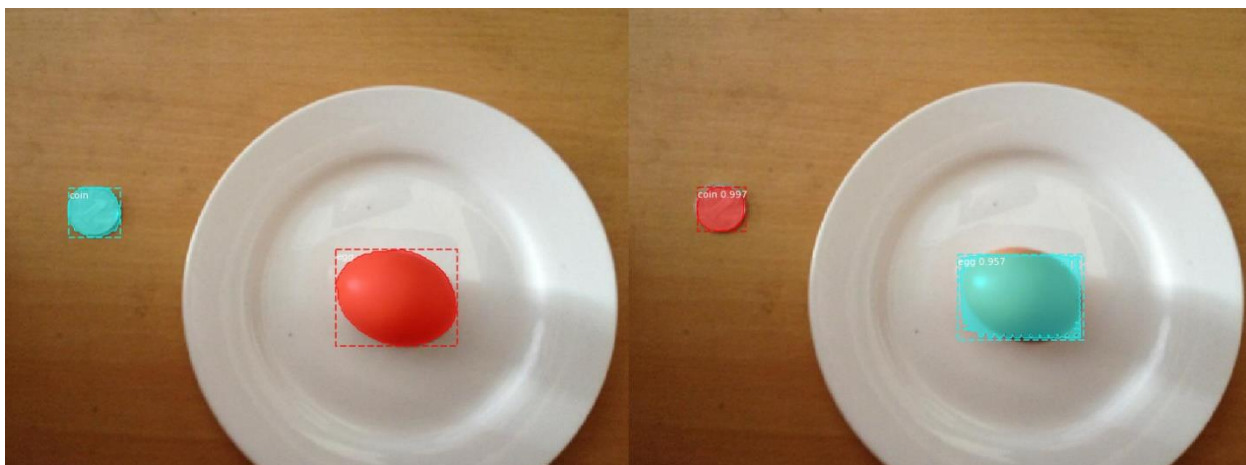
Abu tinklai sugebėjo teisingai klasifikuoti, tačiau ResNet-50 tinklo loss buvo didesnis. ResNet-101 tinklas mokėsi ilgiau, tačiau gavo geresnį rezultatą, todėl toliau bandymuose naudosiu šį ResNet-101 tinklą.

6.2 Duomenų papildymas

Treniruoti konvoliucinį neuroninį tinklą reikia daug duomenų, tačiau daug jų mes neturime. Todėl naudosiuosi duomenų papildymu ir atliksiu tam tikras nuotraukų transformacijas kaip horizontalūs bei vertikalūs nuotraukų apvertimai, skalės keitimas. Duomenų papildymu bandysiu padidinti savo turima duomenų rinkinį ir tuo pačiu pagerinti modelio tikslumą. Bandymas atliktas ant Mask R-CNN tinklo naudojantis ResNet-101 stuburą (backbone). Modelis buvo treniruojamas 15 epochų, treniruojami visi modelio sluoksniai. Iš kiekvienos nuotraukos buvo gautos 3 nuotraukos su skirtingomis transformacijomis.

6.2.1 Duomenų papildymo rezultatai

Ištreniravus tinklą, pavyko gauti 14 paveikslėlyje pavaizduota modelio spėjimą. mAP yra 0.78 – iki šiol geriausias gautas. Loss taip pat gavosi mažiausias – 0.872. Taigi, duomenų



pav. 14 sužymėta nuotrauka kairėje, modelio spėjimas dešinėje

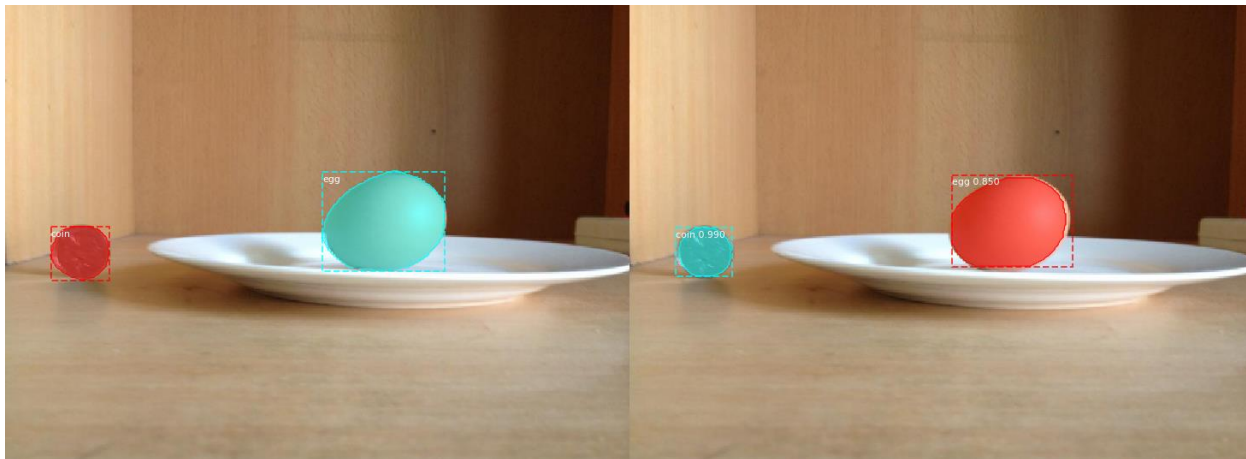
papildymas padėjo modeliui pasiekti geresnių rezultatų. Kadangi modelį pavyko pagerinti, toliau jį mokysime naudodami duomenų papildymą.

6.3 Tikslus derinimas

Tikslus derinimas yra viena iš pernešančio mokymo technikų. Jam reikalingi iš anksto apmokyto modelio svoriai. Ankstesniuose etapuose modelį mokiau nuo pradžių, tačiau šiame etape pasinaudojau MS COCO [COCO] svoriais. MS COCO yra didelio masto objektų aptikimo, segmentavimo bei užrašų duomenų rinkinys. COCO sudarytas iš 330 tūkstančių nuotraukų, 80 objektų kategorijų. Nors COCO duomenų rinkinys neturi man reikalingų maisto produktų, bet šie treniruoti svoriai jau išmoko atpažinti daug bruožų aptinkamų įprastose nuotraukose, todėl tinklas buvo mokytas ne nuo pradžių. Prieš mokindamas modelį užšaldžiau visus tinklo svorius ir leidau mokintis tik paskutiniams. Modelio mokymosi sparta buvo 0.001. Modelį mokinau 15 epochų, taip pat pritaikiau duomenų papildymą, kadangi šis pagerino ankstesnio modelio rezultatus. Duomenis papildžiau atlikdamas horizontalius apvertimus, vertikalius apvertimus, suliejimus, pakeičiau nuotraukos skalę.

6.3.1 Tikslaus derinimo rezultatai

Gavau gana tikslų modelį gebanti atpažinti maistą, nurodyti jo buvimo vietą nuotraukoje dėžutėje bei jo kaukę. 15 paveikslėlyje pavaizduoti gauti segmentacijos rezultatai: nuotraukos kairėje yra mano sužymėtos nuotraukos ir klasės, nuotraukos dešinėje yra Mask R-CNN modelio apmokyto duomenų bazėje klasės ir kaukės spėjimas. Gauti rezultatai buvo geriausi iki šiol gauti. Loss – 0.2359, mAP – 0.95.



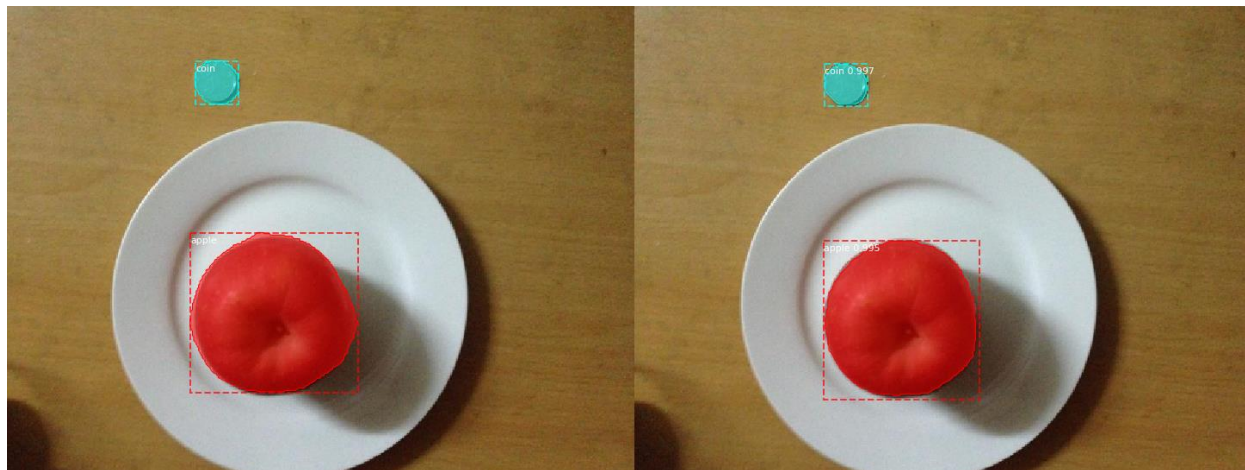
pav. 15 Kairėje mano sužymėtos nuotraukos ir modelio spėjimai

6.4 Tikslus derinimas visam tinklui

Paskutinis Mask R-CNN tinklo apmokymas. Šis bandymas buvo geriausias, tačiau ir mokėsi ilgiausi. Vėl pritaikyta pernešančio mokymo technika – tikslus derinimas. Vėl naudojau MS COCO svorius. Taip pat pritaikiau duomenų papildymą. Duomenis papildžiau atlikdamas horizontalius apvertimus, vertikalius apvertimus, suliejimus, pakeičiau nuotraukos skalę. Šį kartą užšaldžius visą tinklą ir 10 epochų leidau mokytis paskutiniams sluoksniams. Tuomet dar 10 epochų leidau mokytis paskutiniams sluoksniams ir viduriams sluoksniams. Visą laiką mokymo sparta buvo 0.001. Galiausiai, tinklą vėl mokiau 10 epochų, leisdamas mokytis visiems tinklo sluoksniams, tačiau sumažinau mokymosi sparta iki 0.0001.

6.4.1 Tikslaus derinimo visam tinklui rezultatai

Iš viso tinklas mokėsi 30 epochų, o prieš tai buvę tinklai mokėsi tik 15 epochų. Gavau tiksliausią modelį iki šiol. 16 paveikslėlyje pavaizduoti gauti segmentacijos rezultatai: nuotraukos kairėje yra mano sužymėtos nuotraukos ir klasės, nuotraukos dešinėje yra Mask R-CNN modelio apmokyto duomenų bazėje klasės ir kaukės spėjimas. Gauti rezultatai loss – 0.1212, mAP – 0.99375. Iš kaukės matome, kad gauti rezultatai nelabai kuo skiriasi nuo pateiktų duomenų.



pav. 16 Kairėje mano sužymėtos nuotraukos ir modelio spėjimai

6.5 Tyrimo rezultatai

Tyrimo metu ištirti ResNet-50 ir ResNet-101 tinklai su mano sudarytu ir sužymėtu nedideliu duomenų rinkiniu. Tuomet geresniam ResNet-101 tinklui pritaikytas duomenų papildymas, taip stebinti, ar padidinus duomenų kiekį bus gaunami geresni rezultatai. Eksperimentas pavyko, gauti geresni rezultatai. Galiausiai buvo išbandyta pernešančio mokymo technika – tikslus derinimas. Naudojantis MS COCO [COCO] svoriais apmokiau savo modelį. Gavau labai gerus rezultatus. Visi tyrimu rezultatai pavaizduoti 2 lentelėje.

Tyrimo pavadinimas	Loss	mAP
Mask R-CNN ResNet-50	0.9045	0.75
Mask R-CNN ResNet-101	0.8928	0.75
Duomenų papildymas ResNet-101	0.872	0.78
Tikslus derinimas ResNet-101	0.2359	0.95
Tikslus derinimas visam modeliui ResNet-101	0.1212	0.99

Lentelė 2 Tyrimo rezultatai

7 Tūrio apskaičiavimas

Kai jau turime tikslų modelį, gebanti klasifikuoti maistą ir nurodyti jo vietą nuotraukos erdvėje bei nurodyti jo kaukę, galime eiti link kito etapo – tūrio apskaičiavimo. Šiame etape iš nuotraukos kuo tiksliau reikia gauti skaičiuojamo maisto tūrį. Išsaugojau tikslaus derinimo svorius, jais naudosisi bandydamas nustatyti maisto tūrį. Kiekvieną kartą skaičiuojant maisto tūrį, pateikiamos dvi nuotraukos. Viena iš viršaus, kita iš šono. Abi šios nuotraukos turi po monetą kaip kalibruojantį objektą. Abi nuotraukos rodo tą patį maisto produktą. Toliau, pasinaudoju tikslaus derinimo Mask R-CNN modeliu skaičiuodamas tūrį. Tūrio apskaičiavimas vyksta tokia tvarka:

1. Nuotraukoje, darytoje iš viršaus yra moneta. Atpažįstu šią monetą su tikslaus derinimo modeliu, nustatau šios monetos kaukę bei ribojančią dėžutę.
2. Nuotraukoje, darytoje iš viršaus yra maisto produktas. Atpažįstu šį maisto produktą su tikslaus derinimo modeliu, nustatau šio maisto produkto kaukę bei ribojančią dėžutę.
3. Taip pat atpažįstu monetą bei maisto produktą nuotraukoje, darytoje iš šono. Randu šių objektų kaukes ir ribojančias dėžutes.
4. Priklausomai nuo to, koks maisto produktas yra klasifikuotas, taikomi skirtingi skaičiavimai. Lentelėje 3 aprašytos tūrio skaičiavimo metodas priklausomai nuo atpažinto objekto klasės:

Klasė	Tūrio skaičiavimo metodas
Obuolys	Elipsoidas
Bananas	Cilindras
Duonos riekė	Kubas
Virtas kiaušinis	Elipsoidas
Pomidoras	Elipsoidas

Lentelė 3 Pagal maisto tipą jam taikoma formulė skaičiuojant tūrį

7.1 Elipsoidas

Tūriui apskaičiuoti užtenka vienos nuotraukos, darytos iš šono. Randu maisto produkto pikselių skaičių. Žinau monetos tikrą plotą – jį randu iš formulės:

$$S = \pi r^2,$$

kur $r = 1,25\text{cm}$, kadangi monetos diametras 2.5cm . Tuomet randu monetos pikselių skaičių nuotraukoje. Tuomet pasinaudodamas proporcija nustatau tikrą maisto plotą kvadratiniais centimetrais:

$$\text{Tikras maisto plotas} = \frac{\text{Maisto produkto pikselių skaičius} * \text{Tikras monetos plotas}}{\text{Monetos pikselių skaičius}}$$

Iš turimo maisto ploto randu jo r , pavadinkime jį apskaičiuotu spinduliu:

$$\text{Apskaičiuotas spindulys} = \sqrt{\frac{\text{Tikras maisto plotas}}{\pi}}$$

Galiausiai turint apskaičiuota spindulį, galime suskaičiuoti tūrį, ši formulė išplaukia iš rutulio tūrio formulės:

$$Tūris = \frac{4 * \pi * \text{Apskaičiuotas spindulys}^3}{3}$$

7.2 Kubas

Kubo tūris skaičiuojamas iš dviejų nuotraukų – iš viršaus ir šono. Pirma reikia rasti kiek pikselių sudaro monetos diametrą. Monetos diametro ilgis 2.5cm. Randame pikselių į centimetrus daugiklį – pavadinkime jį tiesiog daugikliu:

$$\text{daugiklis} = \frac{2.5}{\text{Monetos diametro pikselių skaičius}}$$

Tuomet randame šoninės nuotraukos maisto aukštį pikseliais ir apskaičiuojame realu aukštį centimetrais:

$$\text{Aukštis} = \text{maisto aukštis pikseliais} * \text{daugiklis}$$

Iš nuotraukos, darytos iš viršaus, randame maisto produkto pikselių skaičių ir monetos pikselių skaičių. Iš proporcijos randame tikrą maisto plotą:

$$\text{Tikras maisto plotas} = \frac{\text{Maisto produkto pikselių skaičius} * \text{Tikras monetos plotas}}{\text{Monetos pikselių skaičius}}$$

Kai jau turime visus parametrus apskaičiuojame tūrį:

$$Tūris = \text{Tikras maisto plotas} * \text{Aukštis}$$

7.3 Cilindras

Cilindro tūriui apskaičiuoti užtenka nuotraukos iš viršaus. Pirma reikia rasti kiek pikselių sudaro monetos diametrą. Monetos diametro ilgis 2.5cm. Randame pikselių į centimetrus daugiklį – pavadinkime jį tiesiog daugikliu:

$$\text{daugiklis} = \frac{2.5}{\text{Monetos diametro pikselių skaičius}}$$

Tuomet randame viršutinės nuotraukos maisto aukštį pikseliais ir apskaičiuojame realu aukštį centimetrais:

$$Aukštis = maisto\ aukštis\ pikseliais * daugiklis$$

Randu maisto produkto pikselių skaičių. Žinau monetos tikrą plotą – jį randu iš formulės:

$$S = \pi r^2,$$

kur $r = 1,25\text{cm}$, kadangi monetos diametras 2.5cm . Tuomet randu monetos pikselių skaičių nuotraukoje. Tuomet pasinaudodamas proporcija nustatau tikrą maisto plotą kvadratiniais centimetrais:

$$\text{Tikras maisto plotas} = \frac{\text{Maisto produkto pikselių skaičius} * \text{Tikras monetos plotas}}{\text{Monetos pikselių skaičius}}$$

Iš turimo maisto ploto randu jo r (naudodamasis cilindro šoninio ploto formule), pavadinkime jį apskaičiuotu spinduliu:

$$\text{Apskaičiuotas spindulys} = \sqrt{\frac{\text{Tikras maisto plotas}}{2 * \text{Aukštis}}}$$

Galiausiai turint apskaičiuota spindulį, galime suskaičiuoti tūrį, ši formulė išplaukia iš cilindro tūrio formulės:

$$Tūris = \pi * \text{Apskaičiuotas spindulys}^2 * \text{Aukštis}$$

7.4 Tūrio skaičiavimo rezultatai

ECUSTFD duomenų rinkinys pateikia nuotraukų testavimui. Kiekvienam objektui buvo suteiktos dvi nuotraukos ir nurodytas tikras jo tūris. Atlikau tūrio skaičiavimus pasinaudojęs anksčiau aprašytais formulėmis. Žemiau, 4 lentelėje pateikti gauti rezultatai, mano apskaičiuotas tūris ir tikras objekto tūris. Iš lentelės matosi, kad yra paklaida skaičiuojant tūrį.

Kai kurių maisto produktu tūris daug nukrypsta nuo tiesos. Taip yra dėl pateiktų fotografijų kampų. Pavyzdžiui, Duonos3 tūris gavosi blogas, nes joje blogai padėta moneta nuotraukoje iš šono. Ji padėta ne viename lygyje su duona, o arčiau kameros, tai matyti paveikslėlyje 17. Taip yra iškraipomos objektų dimensijos ir gaunami prasti rezultatai.

Maisto produktas	Apskaičiuotas tūris	Tikras tūris
Obuolys1	508	310
Obuolys2	258.8	290
Obuolys3	252.6	280
Obuolys4	296.3	300
Obuolys5	340.6	420
Bananas1	127.7	150
Bananas2	135.7	139
Bananas3	158.6	159.5
Bananas4	184.5	153.5
Duona1	123.9	150
Duona2	160	160
Duona3	52	170
Duona4	240	170
Pomidoras1	199.4	210
Pomidoras2	140.2	190
Pomidoras3	150.7	180
Pomidoras4	173.3	160
Kiaušinis1	44.5	50
Kiaušinis2	44.6	50
Kiaušinis3	67	50
Kiaušinis4	61.8	50

Lentelė 4 Suskaičiuotas tūris ir tikras tūris



pav. 17 Duonos3 šoninė nuotrauka

8 Maisto Maistingumo nustatymas

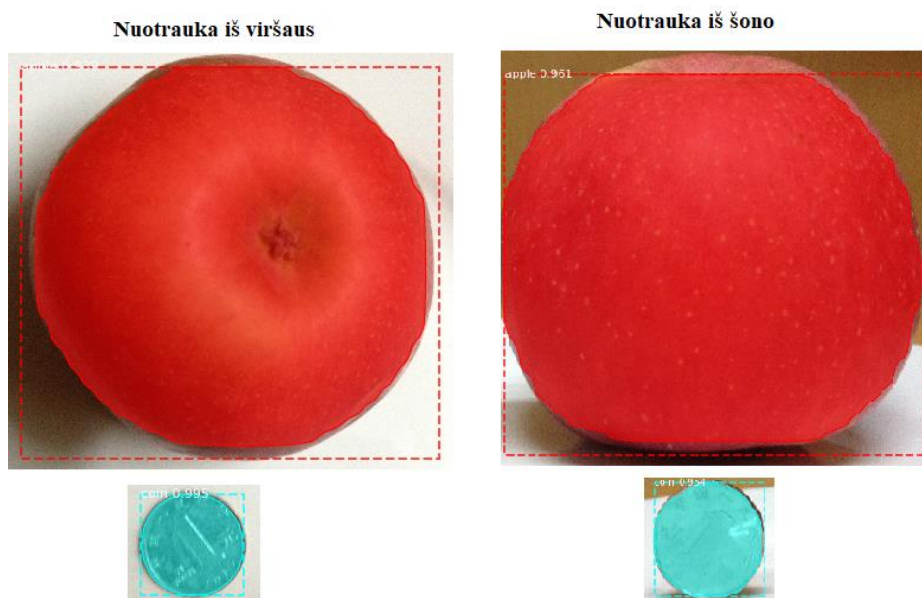
Prieš pradėdant paskutinį etapą, maistingumo nustatymą, dar karta priminsiu prieš tai buvusių etapų nustatant maisto maistingumą:

1 etapas: Įvesties duomenų gavimas. Šiame etape vartotojas pateikia valgomo maisto nuotrauką, darytą iš šono ir viršaus. Taip pat kiekvienoje nuotraukoje turi būti kalibravimo objektas – moneta.



pav. 18 Įvesties duomenis, maisto nuotrauka daryta iš šono ir iš viršaus. Nuotraukose pateikiamas kalibravimo objektas - moneta

2 etapas: Maisto produkto klasifikavimas bei kaukės ir ribojančios dėžutės radimas. Šiame etape naudojamos išreniruotu Mask R-CNN modeliu. Modelis aptinka kiekvieno objekto kaukę, klasę bei ribojančią dėžutę ir pateikia šią informaciją kitam etapui.



pav. 19 Atpažinti objektai bei jų ribojančios dėžutės ir kaukės kiekvienoje nuotraukoje

3 etapas: Tūrio skaičiavimas. Iš 2 etape gautos informacijos naudojantis gilių neuroninių tinklų modelių Mask R-CNN skaičiuojame maisto tūrį. Priklausomai nuo to, koks maistas klasifikuotas, taikomi skirtingi tūrio skaičiavimai. Visa naudojami skaičiavimai pateikti 8 skyriuje.

4 etapas: maisto maistingumo skaičiavimas. Paskutinis etapas. Esame klasifikavę maisto produktą nuotraukoje, taip pat žinome jo tūrį. Dabar beliko apskaičiuoti maisto svorį ir iš jo nustatyti maistingumą. Taip pat reikalingas maisto produkto tankis, kad galima būtų apskaičiuoti jo masę naudojantis formule:

$$Masė = Tankis * Tūris$$

Masė skaičiuojama pagal 5 lentelėje nurodyta maisto produkto tankį:

Maisto produktas	Tankis (g/cm ³)
Obuolys	0.73
Bananas	0.97
Pomidoras	1.1
Duonos riekė	0.2
Virtas kiaušinis	1.14

Lentelė 5 Maisto produktų tankis

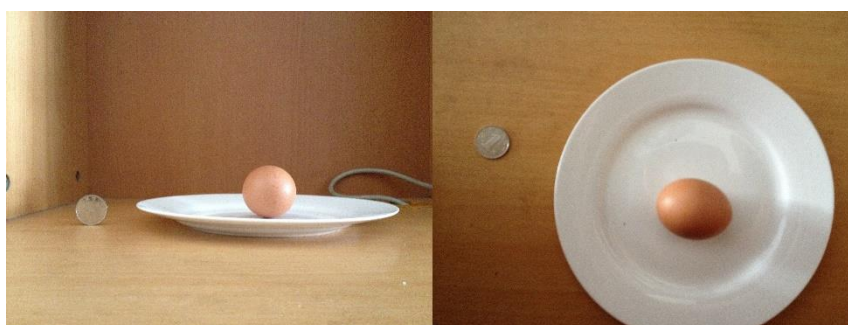
8.1 Maistingumo nustatymo pavyzdys

Šiame skyrelyje pritaikysime išmoktas teorines žinias ir ištreniruotą Mask R-CNN tinklą maisto maistingumo skaičiavime. Pasinaudosiu nemokamai suteikta informacija iš [FoodNutrition] ir susidarysiu paprastą maistingumo lentelę. Dėl tyrimo skaitomumo, nurodysiu tik kai kurias maistingumo vertes. Visos vertės lentelėje nurodytos 1 gramui produkto:

Maisto produktas	Energija (kcal)	Energija (kj)	Angliavandeniai	Cukrus
Obuolys	0.52	2.17	0.14	0.1
Bananas	0.89	3.72	0.23	0.12
Pomidoras	0.18	0.75	0.04	0.024
Duonos riekė	2.66	11.1	0.5	0.06
Virtas kiaušinis	1.56	6.48	0.02	nėra

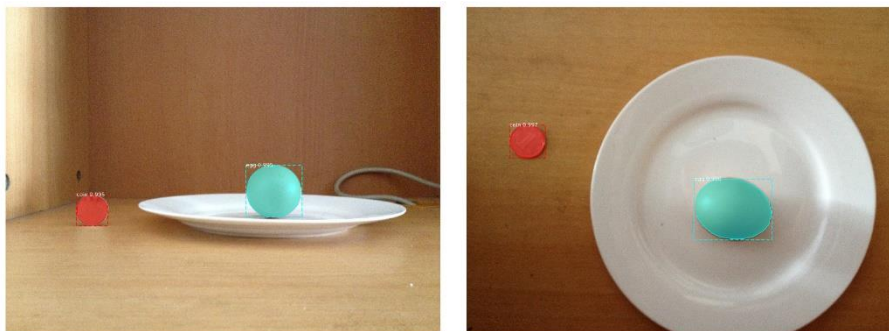
Lentelė 6 Maistingumo lentelė 1 gramui maisto

- Taigi, eikime prie pirmojo etapo, įvesties duomenų. Jiems naudosime Kiaušinis1 nuotrauką:



pav. 20 Įvesties duomenys. Kiaušinio nuotrauka iš šono ir viršaus

- Antrasis etapas. Naudojantis Mask R-CNN modeliu atpažįstu nuotraukose esančius objektus – kiaušinį ir monetą. Taip pat gaunu kiaušinio bei monetos ribojančias dėžutes ir kaukes.



pav. 2111 Mask R-CNN tinklo sugeneruotos kaukės bei ribojančios dėžutės

Gauname, kad maisto klasė yra kiaušinis, taip pat gauname ribojančių dėžučių koordinates ir kaukę – matrica su reikšmėmis 0 ir 1. Ši kaukės matrica nurodo, ar nuotraukos pikselis yra maisto produktas (1) ar nėra (0). Gautas koordinates ir kaukę naudosime trečiame etape – tūrio skaičiavime.

- Trečiasis etapas – tūrio skaičiavimas. Žinome, kad klasifikuotas maisto produktas yra kiaušinis, todėl pritaikome elipsoido tūrio skaičiavimo metodą, kuriam užteks kaukės ir ribojančios dėžutės informacijos iš šoninės nuotraukos:
 1. Iš kiaušinio kaukės apskaičiuojame pikselių skaičių, pažymėtu 1. Mūsų atveju reikšmė lygi 123154.
 2. Iš monetos kaukės apskaičiuojame pikselių skaičių, pažymėtu 1. Reikšmė lygi 39839.
 3. Sudarome proporciją ir gauname maisto kiaušinio plotą kvadratiniais centimetrais. Plotas lygus 15.17 cm^2 .
 4. Iš ploto išsiskaičiuojame spindulį. Spindulys lygus $2.19 \text{ (cm}^2\text{)}$. Turėdami spindulį galime apskaičiuoti tūrį. Tūris lygus $44.46 \text{ (cm}^3\text{)}$.
- Ketvirtasis etapas – maistingumo skaičiavimas.
 5. Randame kiaušinio masę sudaugindami apskaičiuota tūrį su kiaušinio tankiu. Masė lygi 50.68 (g) .
 6. Apskaičiuojame maisto maistingumą sudaugindami kiaušinio masę su lentelėje 6 pateiktomis 1 gramo vertėmis. Gautas kiaušinio maistingumas:

Maisto produktas	Energija (kcal)	Energija (Kj)	Angliavandeniai	Cukrus
Virtas kiaušinis	13.87	57.63	0.18	0

Lentelė 6 Kiaušinio pavaizduoti 19 paveikslėlyje maistingumas

Paimkime iš 8.4 skyriaus dar pora maisto produktų ir apskaičiuokime jų maistingumą:

Maisto produktas	Masė	Energija (kcal)	Energija (Kj)	Angliavandeniai	Cukrus
Obuolys ⁴	216.3	0.52	2.17	0.14	0.1
Bananas ³	153.8	0.89	3.72	0.23	0.12
Pomidoras ¹	219.34	0.18	0.75	0.04	0.024
Duona ²	32	2.66	11.1	0.5	0.06
Kiaušinis ²	50.8	1.56	6.48	0.02	nėra

Lentelė 7 Apskaičiuotas maisto produktų maistingumas

8.2 Rezultatų palyginimas su kitais darbais

Rezultatus lyginsiu su dviem straipsniais. Prieš lyginant suskaičiuosiu absoliutų tikslumą kiekvienai klasei. Šį absoliutų tikslumą skaičiuosiu iš 4 lentelės aprašytų bandymų. Pirmiausia apskaičiuosiu absoliutų tikslumą kiekvienam bandymui ir galiausiai kiekvienam maisto produktui paskaičiuosiu absoliutų tikslumą, kuris bus pateiktas 8 lentelėje. Skaičiavau naudodamasis šiomis formulėmis:

$$\text{Absoliutus tikslumas vienam bandymui} = \frac{\text{Apskaičiuotas tūris} * 100\%}{\text{Tikras tūris}} - 100\%$$

$$\text{Absoliutus tikslumas} = 100\% - \left| \frac{\sum_{N=1}^{\text{Bandymų skaičius}} \text{Absoliutus tikslumas bandymui } N}{\text{Bandymų skaičius}} \right|$$

Maisto maistingumo reikšmė labiausiai priklauso nuo suskaičiuoto tūrio, kadangi maistingumas priklauso nuo suskaičiuotos masės, o masė yra tūris padaugintas iš tankio (konstantos). Dėl šių priežasčių su kitais darbais lyginsiu ne apskaičiuotas kalorijas ar kokią kitą apskaičiuojama vertę, o apskaičiuota tūrį. Pirmiausia iš 4 lentelės skaičiuoto tūrio sudarysiu naują lentelę, kuri rodys tūrio apskaičiavimo tikslumą lyginant su tikru produkto tūriu kiekvienam maisto produktui:

Maisto produktas	Absoliutus tikslumas (%)
Obuolys	95.4
Bananas	99.4
Pomidoras	96.3
Duona	90.2
Virtas kiaušinis	91.9
Vidutinis tikslumas	94.6

Lentelė 8 Absoliutus skaičiavimų tikslumas atliktiems bandymams pavaizduotiems 4 lentelėje

Kai jau turime išsivedę visas reikšmes galime pradėti lyginti rezultatus. Abiejuose lyginamuose darbuose vidutinis tikslumas buvo paskaičiuotas sudedant visus to darbo absoliučius tikslumus kiekvienai maisto klasei ir padalintas iš maisto produktų skaičiaus tame darbe. Pirmą lyginsiu su Parisa Pouladzadeh, Shervin Shirmohammadi autorių straipsniu [MCFFI], kurie maisto

kalorijų skaičiavimui naudojo atraminių vektorių klasifikatorių. Atraminių vektorių klasifikatorius - sistemos mokymosi algoritmas skirtas klasifikuoti duomenims. Tai prižiūrimo mokymosi metodas, kuomet siekiama suklasifikuoti jau pažymėtus duomenis. Taip pat, kaip kalibravimo objektą šiame darbe jie naudojo nykštį. Jų darbe gautos absoliutaus tikslumo reikšmės sutampančioms maisto klasėms pateiktos 9 lentelėje:

Maisto produktas	Absoliutus tikslumas (%)
Obuolys	87
Bananas	89
Pomidoras	76
Duona	76
Virtas kiaušinis	88
Vidutinis tikslumas	86

Lentelė 9[MCFFI] tyrimo absoliutūs tikslumai

Kaip matome, mano gauti rezultatai buvo žymiai geresni lyginti su [MCFFI] darbu. Kiekvienos maisto klasės absoliutus tikslumas buvo geresnis, o vidutinis tikslumas buvo net 8.6% geresnis.

Antras straipsnis su kuriuo lyginsiu savo darbą yra parašytas Sable Vinayak Machhindra [FCEUIP]. Šiame darbe autorius maisto kalorijas skaičiuoja naudodamasis konvoliuciniais neuroniniais tinklais ir Graph Cut algoritmu, taip pat naudodamas rankos pirštą kaip kalibravimo objektą. Šiame darbe konvoliuciniu neuroniniu tinklu atpažįstamas maisto produktas, tuomet pasinaudojus Graph Cut algoritmu, randamas maisto ir nykščio kontūras ir iš jo išskaičiuojamas maisto tūris. Šio darbo sutampančių maisto klasių absoliutus tikslumas pateiktas 10 lentelėje:

Maisto produktas	Absoliutus tikslumas (%)
Obuolys	74.9
Bananas	89.4
Pomidoras	79.2
Vidutinis tikslumas	82.7

Lentelė 10 [FCEUIP] tyrimo absoliutūs tikslumai

Mano rezultatai geresni už šio autoriaus darbą. Kiekvienos sutampančios maisto klasės absoliutus tikslumas buvo tikslesnis, o vidutinis tikslumas buvo 11.7% tikslesnis.

Išvados

Bakalaurinio darbo metu buvo atlikta konvoliucinių tinklų analizė. Išanalizuotas pernešantis mokymas. Ištirtas Mask R-CNN tinklas. Sudarytas ir sužymėtas maisto duomenų rinkinys. Išnagrinėtas tūrio skaičiavimas iš nuotraukų. Buvo aptreniruotas Mask R-CNN tinklas, tuomet pritaikytas duomenų papildymas bandant jį pagerinti. Tinklo tikslumas padidėjo. Tinklui pritaikytas pernešantis mokymas naudojantis MS COCO svoriais bei duomenų papildymas. Tinklas tapo labai tikslus. Gavus labai tikslių tinklą, buvo skaičiuojamas klasifikuoto maisto tūris iš tinklo sugeneruotų kaukių bei ribojančių dėžučių. Skaičiavimams buvo pritaikytos teorinės tūrio skaičiavimo žinios. Gautas vidutinis tūrio tikslumas buvo 94.6%. Gautas modelis palygintas su dviem straipsniais, kuriuose taip pat skaičiuojamas maisto maistingumas. Viena iš šių straipsnių autoriai naudoja konvoliucinius neuroninius tinklus, kitame straipsnyje naudojamas atraminių vektorių klasifikatorius. Mano tinklo gauti rezultatai žymiai geresni už šių abiejų tinklų rezultatus.

Galutinis darbo tikslas – sukurtas algoritmas, kuriam nereikia jokios išorinės informacijos ar programinės įrangos, tik vartotojo įvesties duomenų (maisto nuotraukų). Šis algoritmas tiksliai apskaičiuoja nufotografuoto maisto maistingumą. Šis algoritmas pirmasis toks, kuris naudoja Mask R-CNN modelį skaičiuodamas maisto maistingumą. Prieš tai buvę algoritmai, kurie maistingumą skaičiuodavo konvoliuciniais neuroniniais tinklais, naudojami papildomais algoritmais ar programomis skaičiuodamos maisto tūrį iš nuotraukų. Taip pat, bakalauriniame darbe naudojama vienu iš naujausių kompiuterių regos modelių Mask R-CNN.

Prieinu prie išvados, kad bakalaurinis darbas pavyko. Sukurtas algoritmas, kuris naudodamasis giliais konvoliuciniais neuroniniais tinklais iš nuotraukų sugeba tiksliai nustatyti maisto maistingumą.

Šaltiniai

- [MASKRCNN] Mask R-CNN, Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick, 2017, <https://arxiv.org/abs/1703.06870>
- [FPN] Feature Pyramid Networks for Object Detection Tsung-Yi Lin¹, Piotr Dollar , Ross Girshick, Kaiming He , Bharath Hariharan, and Serge Belongie , http://openaccess.thecvf.com/content_cvpr_2017/papers/Lin_Feature_Pyramid_Networks_CVPR_2017_paper.pdf
- [FASTERRCNN] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, <https://arxiv.org/abs/1506.01497>
- [Matterport] W. Abdulla, Mask r-cnn for object detection and instance segmentation on keras and tensorflow, https://github.com/matterport/Mask_RCNN (2017).
- [ECUSTFD] Computer vision-based food calorie estimation: dataset, method, and experiment, Liang, Yanchao and Li, Jianhua, 2017
- [COCO] Microsoft COCO: Common Objects in Context, Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár, 2014
- [FoodNutrition] https://www.canada.ca/content/dam/hc-sc/migration/hc-sc/fn-an/alt_formats/pdf/nutrition/fiche-nutri-data/nvscf-vnqau-eng.pdf
- [ASTL] A Survey on Transfer Learning Sinno Jialin Pan and Qiang Yang Fellow, IEEE, https://www.cse.ust.hk/~qyang/Docs/2009/tkde_transfer_learning.pdf
- [NU17] Simon Mezgec ¹ and Barbara Koroušić Seljak ². NutriNet: A Deep Learning Food and Drink Image Recognition System for Dietary Assessment, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5537777/>
- [MCFFI] Measuring Calorie and Nutrition From Food Image Parisa Pouladzadeh, Shervin Shirmohammadi, Senior Member, IEEE, and Rana Al-Maghrabi, <https://cutt.ly/dybJ4Uu>
- [FCEUIP] Sable Vinaysa Machhindra, Foods Calories Estimation Using Image Processing, <https://cutt.ly/pybJ38J>

Priedai

1 priedas. Maisto maistingumo skaičiavimas

Nuoroda, kurioje patalpintas algoritmas maisto maistingo skaičiavimui:

<https://www.kaggle.com/vytasmatas/calculate-foods-nutrition/>

2 priedas. Modelio treniravimas

Nuoroda, kurioje patalpintas algoritmas Mask R-CNN modelio treniravimui bei geriausiems modelio svoriams gauti:

<https://www.kaggle.com/vytasmatas/train-model>