VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Baigiamasis bakalauro darbas

Tatuiruočių aptikimas naudojantis konvoliuciniais neuroniniais tinklais

(Tattoo detection using Convolutional Neural Network)

Atliko: 4 kurso, 2 grupės studentas
Saulius Karmanovas (parašas)
Darbo vadovas:
Andrius Grevys (parašas)
Recenzentas:
prof. dr. Rimantas Vaicekauskas (parašas)

Vilnius
2019

Turinys

Įvadas 3
1. Vaizdų atpažinimo problema 4
2. Nuotraukų formatas 4
3. Neuroniniai tinklai 5
3.1. Pilnai sujungti sluoksniai 6
3.2. Konvoliuciniai sluoksniai 6
3.3. Maksimumų arba vidurkių telkimo sluoksniai 7
4. Konvoliucinių neuroninių tinklų objektų aptikimui architektūros 8
4.1. Bruožus formuojantys tinklai 9
4.1.1. ResNet 9
4.2. Tinklų architektūros 10
4.2.1. R-FCN 10
5. Konvoliucinių neuroninių tinklų mokymas 11
6. Duomenų bazės sudarymas 12
7. Duomenų bazės modifikavimas 13
7.1. Afininės transformacijos 14
7.2. Kelių pavyzdžių suliejimas 15
8. Neuroninio tinklo žinių perdavimas 16
9. Bandymai 17
9.1. Pavyzdžių suliejimo rezultatai 18
9.2. Žinių perdavimo rezultatai 19
9.3. PR kreivės 21
10. Pavyzdžiai 22
11. Išvados 25
12. Summary 25

Įvadas

Su kiekviena nauja vaizdo plokščių karta neuroninių tinklų prieinamumas vis didėja, o kartu tobulėjant vaizdų atpažinimo technologijoms, galima vis įvairesnėms sritims pritaikyti kompiuterių regos sistemas. Žmonių atpažinimas yra viena dažnai sutinkama užduotis kompiuterių regos srityje. Tačiau žmonių išvaizda nėra pastovi, o ją pakeisti yra nesudėtinga. Užtat tatuiruotės yra ilgalaikės, metams bėgant beveik nekinta, o jų pašalinimas brangus ir sudėtingas procesas. Tatuiruotės puikiai tinka žmonių identifikavimui, nes yra pastovios, ir dažnai unikalios. Kaip ir kiti meno kūriniai, tatuiruotes nuo kopijavimo saugo įvairios autorinės teisės. Dėl šių teisių pažeidimo retai atsiranda bylų, tačiau svetimų tatuiruočių kopijavimo meistrai beveik nesiima. [Per13] Visos šios priežastys puikiai pavaizduoja, kodėl kriminalistikoje ir kitose srityse tatuiruotės yra ypač naudingos identifikuojant žmones. Pagal jas įmanoma nustatyti žmogaus priklausymą nusikalstamoms grupuotėms bei įvairiom gaujom. Siekiant šį procesą perkelti į kompiuterį ir automatizuoti pakartojamumui reikalinga sistema, kuri sugebėtų atlikti dvi svarbias užduotis: atrasti vaizdo įraše, o paprasčiau nuotraukoje, tatuiruotes bei sugebėti kelias iš jų palyginti ir nustatyti ar dviejose nuotraukose yra užfiksuota ta pati tatuiruotė. Tik pirmoji užduotis bus nagrinėjama šiame darbe. Pagrindinis šio darbo tikslas buvo ištirti, kaip modernios architektūros konvoliucinis neuroninis tinklas sugeba prisitaikyti prie poros skirtingų mokymo būdų sprendžiant tatuiruočių aptikimo problemą. Šiam tikslui pasiekti buvo pasirinktas R-FCN atpažinimo tinklų modelis, mokymui naudota surinkta tatuiruočių duomenų bazė, apmokyti du modeliai naudojant pavyzdžių suliejimą bei keletas mažesnių modelių pasitelkus žinių perdavimą bei atlikta jų analizė.

1. Vaizdų atpažinimo problema

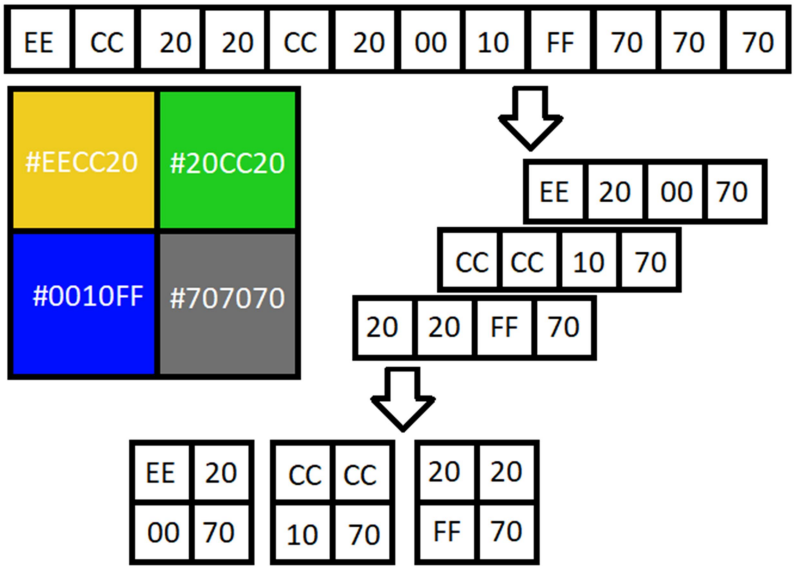
Vaizdų atpažinimas yra viena iš kompiuterinės regos sričių, kurią nagrinėjant susiduriama su pagrindine problema, kuri yra reikalingos informacijos didelis kiekis bei įvairumas. Kad ir kokiu būdu yra sprendžiamas šis uždavinys, siekiant įveikti šią kliūtį naudojama sistema (nesvarbu ar tai neuroninis tinklas ar euristinis algoritmas) turi būti pakankamai sofistikuota, kad sugebėtų prisitaikyti prie tokių duomenų variacijų paveikslėliuose:

- aptinkami daiktai gali būti pavaizduoti įvairiomis orientacijomis, skirtingose nuotraukos srityse, įvairiais dydžiais;
- vien iš nuotraukos gali būti sunku nustatyti objekto formas ar ribas dėl to, kad yra prarandama naudinga erdvinė informacija, kai šie trimačiai objektai atvaizduojami į plokštumą;
- nevienoda arba bloga aplinka bei apšvietimas, kuriuose nufotografuoti tiriami objektai;
- daiktai gali būti uždengti kitų tiriamų objektų bei aplinkos arba dalinai išlindę už paveikslėlio ribų;
- foninis triukšmas arba dažnai pasikartojanti nebūtina informacija (pvz. lėktuvą turi atskirti nuo kitų daiktų ne pagal tai, kad jie dažnai pavaizduoti šviesiame mėlyname fone);
- vienos klasės objektų įvairovė (pvz. mašinų klasei gali priklausyti skirtingų spalvų, modelių bei dydžių automobilių, kuriuos visus sistema turi išmokti atpažinti ir priskirti vienai klasei);
- objektų panašumai tarp skirtingų klasių (pvz. nors tatuiruotės nuotraukoje yra vizualiai panašios į užrašus ant rūbų ar pastatų, tačiau tik jos priklauso tatuiruočių klasei);

Be abejo, kad sistema išmoktų atpažinti daiktus nuotraukose, pirma reikia nurodyti kompiuteriui suprantamu būdu, kaip bus aprašomos nuotraukos, iš kurių mokysis aptikti ir klasifikuoti objektus.

2. Nuotraukų formatas

Paveikslėliams saugoti naudojama daug skirtingų failų formatų, tačiau visus juos galima užkrauti į kompiuterio atmintį kaip nuoseklią baitų seką. Tokioje sekoje vienas baitas nusako raudonos, žalios arba mėlynos spalvos stipruma (trijų spalvų kanalų atveju) viename nuotraukos taške. Tačiau sistemą mokyti naudojant tokią vientisą seką nėra patogu, nes šiuo būdu yra prarandama informacija apie gretimus plokštumos taškus. Nuotraukoje gretimi taškai vienam pikseliui šioje vientisoje sekoje yra atskiriami vienos paveikslėlio taškų eilutės duomenimis. Dėl šios priežasties nesudėtingomis transformacijomis galima sudaryti naują nuotrauką nusakančia struktūrą, kuri išlaiko dvimatę duomenų erdvę. Kiekvieną spalvų kanalą atitinka kas trečias sekos baitas, todėl išrinkus kiekvieno kanalo duomenis galima sudaryti tris naujas sekas, kuriose du horizontaliai iš eilės einantys taškai atitiks sekoje gretimus baitus. Šias tris ilgas baitų sekas (sluoksnius) suskirsčius periodiškai pagal paveikslėlio eilutes gaunamos trys matricos, kuriose kiekvienas elementas a_{ij} atitinka vieną i-tosios nuotraukos eilutės j-tąjį tašką ekrane (pav. 1).



1 pav. Kaip keturių taškų nuotrauką yra paverčiama į trijų spalvų kanalų matricas.

Tokios struktūros duomenimis manipuliuoja konvoliuciniai tinklo sluoksniai. Jie formuoja lokalią informaciją apie nuotrauką (aptinka kampus, briaunas, reljefo pokyčius). Kartu tokia duomenų struktūra leidžia nesudėtingai „iškirpti“ stačiakampius regionus, kurių ieškoma sprendžiant šią objektų aptikimo užduotį.

3. Neuroniniai tinklai

Pastaruoju metu kompiuterių regos srityje dažnai nagrinėjami konvoliuciniai tinklai yra stipriai pasikeitę nuo pirmųjų neuroninių tinklų, tačiau net juose vis dar galima įžvelgti panašumų į biologinius neuroninius tinklus. Tiek dabartiniai, tiek pirmaisiais neuroniniais tinklais yra siekiama imituoti biologinių neuroninių tinklų informacijos ir vaizdų apdorojimo būdą, kuris išsivystė evoliucijos pagalba. Abiejų rūšių tinklai susideda iš mažų paprastų dalialyčių, vadinamų neuronais, kurie susideda iš signalą priimančios dalies, signalą apdorojančios dalies bei kitam neuronui signalą perduodančios dalies. Šiuolaikiniuose neuroniniuose tinkluose šie neuronai yra grupuojami sluoksniais, kuriuose tame pat sluoksnyje esantys neuronai atlieka vienodą funkciją. Tą pačią funkciją atliekantys neuronai tarpusavyje skiriasi tik vidiniais signalo modifikacijos svoriais, kurie nusako, kaip keičiasi signalas.

Šie neuronų sluoksniai sistemoje yra sujungti kryptinga tvarka, kad informacija (signalas) sklistu tinklu gilyn. Tai yra, signalas patekęs į pirmą sluoksnį yra modifikuojamas ir rezultatas perduodamas į antrąjį sluoksnį, iš jo į trečią ir taip toliau. Pasitelkus daug tokių primityvių, sujungtų sluoksnių, gautą neuroninį tinklą galima išmokyti, keičiant vidinius svorius, imituoti įvairių sudėtingų funkcijų veikimą. [LBH15] Tokiu būdu per apmokytą tinklą keliavusi ir pabaigoje išėjusi informacija yra pradinius duomenis atitinkantis uždavinio atsakymas. Objektų aptikimo uždavinyje toks rezultatas būtų regionų sąrašas (objektus glaudžiau apgaubiančių stačiakampių kraštinių ribos) bei šių regionų klasifikavimo įverčiai (, kur didžiausios reikšmės indeksas nusako, kuriai klasei labiausiai tikėtina, kad priklauso aptiktas objektas).

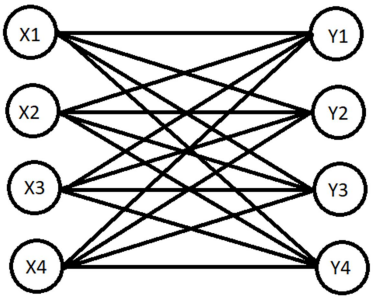
Dažniausiai sutinkami neuroninių tinklų sluoksniai yra suskirstomi į tokias grupes pagal jų naudojamas funkcijas:

- pilnai sujungti sluoksniai;
- konvoliuciniai sluoksniai;

- maksimumų arba vidurkių telkimo sluoksniai;

3.1. Pilnai sujungti sluoksniai

Pilnai sujungti sluoksniai yra paprasčiausia svorinė įvesties suma kiekvienam sluoksnio rezultato elementui. Tačiau, kai net ir nedidelių tinklų įvesties sluoksniai nesunkiai viršija keliasdešimt tūkstančių kintamųjų [KSH12], kiekvienam sluoksnio išvesties elementui naudojami kiti svoriai, o dėl tokio svorių kiekio šiuos sluoksnius optimizuoti yra sudėtinga. Į sluoksnį patekę duomenys laikomi vektoriuje, kuris yra sudauginamas su sluoksnio svorių matricą bei pridedamas papildomas vienas svoris (pav. 2). Šiuo būdu kiekvieną išeinančią jungtį galima išreikšti visų įėjusių duomenų svorine kombinacija. Šiame sluoksnyje reikia optimizuoti visas svorių matricos reiškmes, o dėl šios priežasties tokių sluoksnių skaičiavimų sudėtingumai greit auga ir juos tampa sudėtinga optimizuoti.



$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} \\ w_{21} & w_{22} & w_{23} & w_{24} & w_{24} \\ w_{31} & w_{32} & w_{33} & w_{34} & w_{35} \\ w_{41} & w_{42} & w_{43} & w_{44} & w_{45} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ 1 \end{pmatrix}$$

2 pav. Kairėje - pilnai sujungtas sluoksnis su keturiomis įvedimo jungtimis (X) ir keturiom išeinančiom jungtimis (Y); dešinėje - pilnai sujungtą sluoksnį atitinkanti matricų daugybą su svoriais W.

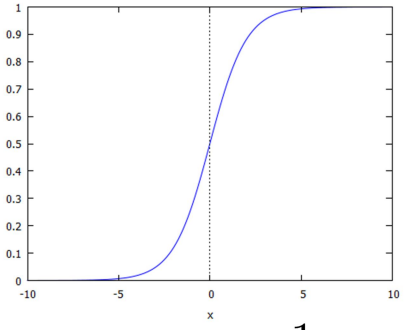
3.2. Konvoliuciniai sluoksniai

Konvoliuciniai sluoksniai yra sudaryti iš daug skirtingų nedidelių filtrų. Duomenys patenka į šį sluoksnį trijų dimensijų struktūroje. Dvi iš šių dimensijų atstoja nuotraukos ilgį ir plotį, o trečia dimensija - prieš tai buvusio sluoksnio filtrų skaičius. Tais atvejai, kai konvoliucinis sluoksnis yra pats pirmas neuroninio tinklo sluoksnis, tada įvesties plokštuma yra nuotraukos dimensijos, o gylis atitinka spalvų kanalai. Pačio konvoliucinio filtro struktūra sudaryta vadovaujantis principu, kad lokali informacija yra svarbesnė kompiuterių regoje. Filtro ilgis ir plotis yra nedideli ir beveik visada vienodi - dažniausiai po 3 arba po 5, o filtro gylis yra tokio pat kaip ir įvestiesties gylis. Šis filtras juda per visą įvesties plokštumą ir kiekviename jos taške apskaičiuoja naują išvesties rezultatą. Jis yra gaunamas filtro svorius padauginus iš atitinkamų nuotraukos taškų, gretimų pasirinktam pikseliui. Tais atvejais, kai kažkuri filtro dalis atsiduria už įvesties ribų, tuomet naudojamas nulinis apvalkalas (zero padding), kuris užtikrina, kad visi filtro apimti elementai už nuotraukos ribų būtų skaitomi, kaip nuliai. Be to, toks apvalkalas padeda išlaikyti nepakitusias plokštumos dimensijas, tai yra, kad po kiekvieno sluoksnio plokštumos kraštai nesitrauktų. Be to filtrai gali skirtis žingsnio dydžiais. Nuo jo priklauso kelintas elementas bus pasirinktas sekantis, tai yra pasirinkus žingsnio dydi 2, filtras sustos kas antrame elemente, kad paskaičiuotų išvesties rezultatą, be to su tokiu filtru sluoksnio išvesties plokštuma sumažėja keturis kartus. Didinant žingsnį būtina atsižvelgti į filtro plotą, kad kiekvienas įvesties elementas būtų apimtas bent vieno filtro, kad nebūtų prarasta informacija. Filtrų skaičius atitinka sluoksnio išeities gylį, nes kiekvienas iš filtrų sukuria po vieną naują išeities plokštumą. Pvz. Jeigu įeinančios struktūros dimensijos 13x13x3 ir naudojami penki 3x3x3 filtrai su žingsniu 2, tada iš tokio sluoksnio išeinanti struktūra turi dimensijas 7x7x5 (pav. 3)

$$(w,\ h,\ d),\ (c,\ x,\ y,\ d,\ s) \Rightarrow \left(\left\lceil \frac{w}{s} \right\rceil,\ \left\lceil \frac{h}{s} \right\rceil,\ c\right)$$

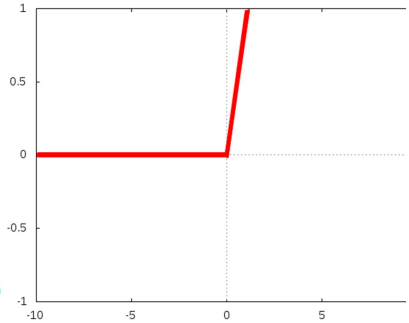
3 pav. Formulė skirta apskaičiuoti kovoliucinio sluoksnio išeities dimensijas. w, h, d - įėjimo ilgis, plotis, gylis; c - filtrų kiekis; x, y, d, s atitinkamai filtro ilgis, plotis, gylis, žingsnis. Gaunamas išėjimo ilgis, plotis ir gylis.

Atlikus kiekvieną konvoliuciją rezultatui dažnai pritaikoma aktyvacijos funkcija. Šis sluoksnis neturi jokių mokymo metu besikeičiančių kintamųjų, jis visus gautus elementus praleidžia per visad tokią pat funkciją ir atiduoda gilesniam sluoksniui. Dažniausiai naudojamos aktivacijos funkcijos: sigmoid (pav. 4), ReLU (pav. 5), leaky ReLU (pav. 6).



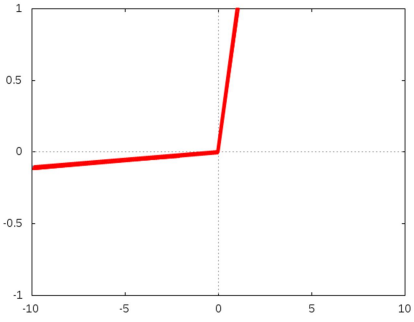
$$f(x) = \frac{1}{1+e^{-x}}$$

4 pav. sigmoid



$$f(x) = \begin{cases} x, & \text{jeigu } x \geq 0 \\ 0, & \text{jeigu } x < 0 \end{cases}$$

5 pav. ReLU



$$f(x) = \begin{cases} x, & \text{jeigu } x \geq 0 \\ 0.01x, & \text{jeigu } x < 0 \end{cases}$$

6 pav. Leaky ReLU

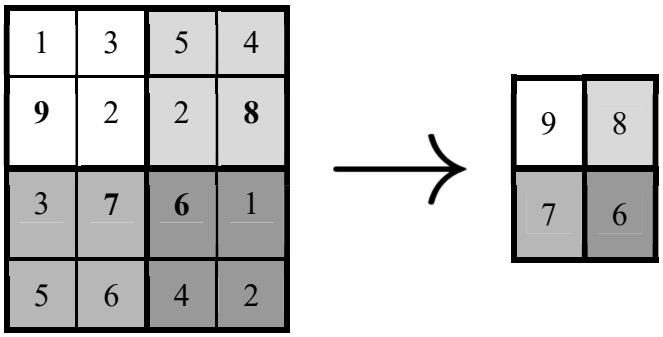
Naudojamų kintamųjų skaičių konvoliuciniame sluoksnyje galima rasti naudojant formulę:

$$|W| = c * x * y * d$$

c - filtrų kiekis; x, y, d atitinkamai filtro ilgis, plotis, gylis.

3.3. Maksimumų arba vidurkių telkimo sluoksniai

Maksimumų (arba vidurkių) telikimo sluoksniai yra panašūs į konvoliucinius sluoksnius tuo, kad jie irgi remiasi lokalia informacija skaičiuojant rezultatus, tačiau kiekvienai plokštumai įvedimo gylyje išvesties rezultatai yra skaičiuojami atskirai. Kaip ir konvoliuciniame sluoksnyje telkimo dažniausiai filtras yra kvadratinis ir visad naudojamas žingsnio dydis. Pagrindinio šio sluoksnio paskirtis yra sumažinti duomenų kiekį, kad tolimesniems sluoksniams reikėtų mažiau skirti skaičiavimų smulkiems bruožams aptikti ir daugiau bendresniems giliasniems požymiams atpažinti. Tokiu būdu kartu padidėja modelio atsparumas nereikalingai informacijai bei vizualiajam triukšmui [BPL10]. Dėl šios priežasties kaupimo sluoksniai sumažina išvesties plokštumą, o šią supaprastintą plokštumą galima daug greičiau apdoroti įvairiais papildomais konvoliuciniais filtrais gilesniuose sluoksniuose. Dažniausiai naudojami kaupimo filtrai yra persidengiantys 3x3 su žingsniu 2 arba nepersidengiantys 2x2 su žingsniu 2 (pav. 7). Kaupimo sluoksniai su didesniais filtrais yra nenaudojami, nes jie yra per daug destruktyvūs ir per greitai atmeta daug naudingos informacijos. Šie filtrai kaip ir konvoliuciniame sluoksnyje po truputį slenka per įvesties plokštuma pasirinktu žingsniu ir kiekviename regione kuriame sustoja pasirenkamas maksimumas (arba vidurkių telkimo sluoksnyje - apskaičiuojamas elementų vidurkis) iš visų filtro uždengtų elementų.



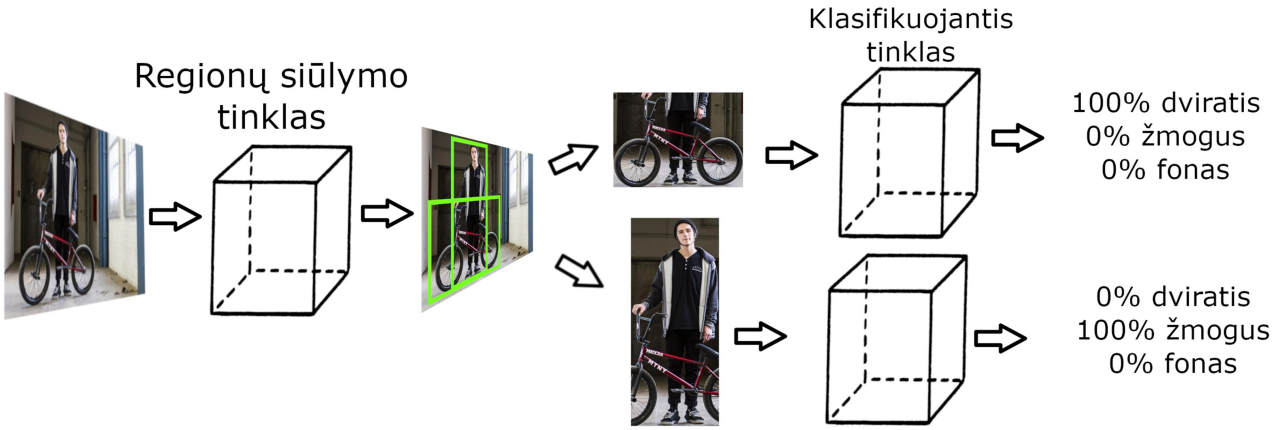
7 pav. 4x4 sluoksnį sumažina iki 2x2 dydžio, iš kiekvieno ketvirčio pasirenkant didžiausią reikšmę.

Telkimas neturi jokių kintamųjų, kuriuos reikėtų optimizuoti mokymosi metu, todėl šie sluoksniai atlieka nesudėtingą operaciją, kuri padeda sumažinti mokymosi sudėtingumą gilesniems sluoksniams.

4. Konvoliucinių neuroninių tinklų objektų aptikimui architektūros

Architektūras dažniausiai sutinkamas objektų aptikimo srityje galima padalinti į dvi grupes: vientiso tinklo architektūros ir dviejų dalinių tinklų architektūros. Vientiso tinklo architektūros modeliai visą objektų aptikimo užduotį sprendžia vienu kartu ir dažnai pasižymi tuo kad veikia greičiau negu dviejų dalių tinklai.

Šiame darbe yra nagrinėjami tik dviejų dalinių tinklų architektūrų tinklai. Šios architektūros modeliai yra paprastesni veikimo atžvilgiu, nes juos sudaro du subtinklai, kurie atlieka skirtingas užduotis. Jie abu kaip pagrindą naudoja tą patį pradinį bruožus formuojantį tinklą, todėl mokant spręsti užduotį nereikia apmokyti dviejų didelių atskirų tinklų. Pirmasis subtinklas sprendžia regionų siūlymo užduotį, kurios tikslas greitai aptikti regionus kurie galimai apima objektą. Bruožus formuojančio tinklo rezultatai kartu su pasiūlytai regionais patenka į klasifikavimo užduotį sprendžiantį subtinklą. Ši užduotis yra primityvesnė nei pats objektų aptikimas, nes jos tikslas yra tik pasakyti kokios klasės objektas yra jau duotame nuotraukos regione (pav. 8).



8 pav. Regionų siūlymo modulis sugeneruoja aibę regionų, kuriuos perduoda klasifikuojančiam moduliui, kuris kiekvienam regionui priskiria vieną iš objektų klasių arba fono klasę.

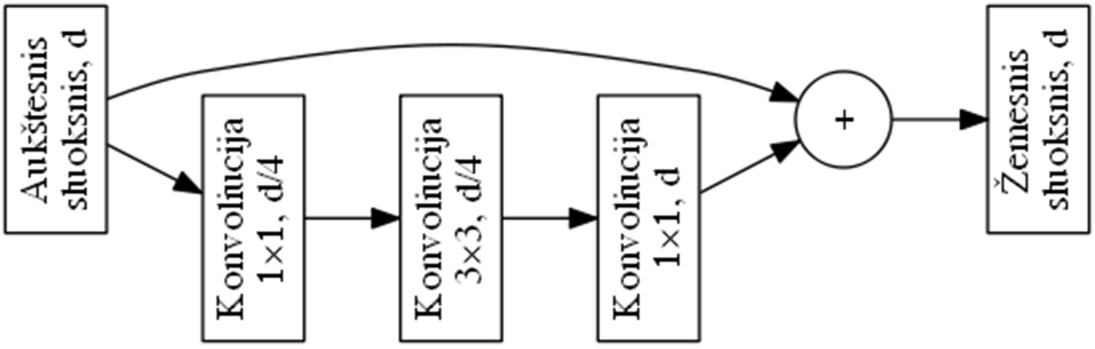
4.1. Bruožus formuojantys tinklai

Dauguma kompiuterių regos užduočių sprendžiančių modelių yra panašūs tuo, kad savo užduotį sprendžia pirma apdorojant pradinę nuotrauką į užduočiai svarbius bruožus. Modeliai, kurių paskirtis yra šių bruožų gavimas iš pradinių duomenų vadinami bruožus formuojančiais tinklais (angl. “feature extractor”). Nors šie bruožai ir juos sukuriantys modelio svoriai nėra vienodi visoms užduotims, tačiau jie yra daug pranašesni už atsitiktinius duomenis, kaip naujos užduoties modelio mokymo pradžios taškas. Dėl šios priežasties dažnai nereikia viso tinklo mokyti nuo nulio, o galima panaudoti dalį panašią užduotį sprendžiančio modelio. Be to, jau dalinai apmokytam naujam modeliui nereikia tiek daug duomenų mokymo metu. Daugumai kompiuterių regos užduočių modelių kaip pagrindas naudojami klasifikavimo modeliai. Vaizdų klasifikavimo užduotis lyginant su kitomis kompiuterių regos užduotimis nėra sudėtinga bei jai reikalingų duomenų rinkimas yra paprastas. Dėl šios priežasties dauguma modelių pirma yra apmokomi naudojant jau surinktas dideles aibes klasifikavimo duomenų ir tik po to prie apmokyto modelio prijungiamos kitos tinklo dalys skirtos specifiniai užduočiai spręsti. Šie bruožus formuojantys tinklai skaičiavimų požiūriu yra sudėtingiausia viso objektų aptikimo modelio dalis, nes atlieka didžiąją dalį visų skaičiavimų, todėl nuo pasirinkto subtinklo stipriai priklauso viso modelio kintamųjų kiekis / tinklo dydis, greitis, tikslumas bei duomenų bazės reikalingos mokymui dydis.

Norint apmokytą klasifikavimo užduotį spręsti modelį pritaikyti objektų aptikimo užduočiai spręsti yra pašalinami klasifikuojančio modelio keli paskutiniai sluoksniai ir prie gauto tinklo prijungiamas iš suformuotų bruožų objektus aptikantis tinklas.

4.1.1. ResNet

Šiame darbe naudoti keli ResNet architektūros bruožus formuojantys tinklai (ResNet 101 ir ResNet 50), kurie skiriasi tik konvoliucinių blokų skaičiumi. Šios architektūros tinklai yra gan gilūs, tačiau sudaryti iš blokų, kurie turi šaką, kuri apeina bloko konvoliucijas ir sudedama su rezultatu (pav.9). Ši savybė modeliui padeda išlaikyti stabilų mokymasi bei sumažinti mokymo trukmę. Be šios šakos blokuose didinant tinklo gylį tikslumas pradeda didėti vis lėčiau arba net mažėja, nes optimizavimo metu išvestinės keliaujančios per tinklą dažnai išnyksta iki 0 arba sprogsta (greitai išauga iki begalybės) [HZR+15]. ResNet tinklo blokai kartu be apeinačios šakos turi šaką sudarytą iš trijų konvoliucinių sluoksnių, kur pirmo bloke sluoksnio 1×1 konvoliucija sumažina gylį, kad sekančiai 3×3 konvoliucijai reiktų atlikti mažiau skaičiavimų, ir po to trečiame bloko sluoksnyje 1×1 konvoliucija duomenis grąžina prie pradinio gylio.

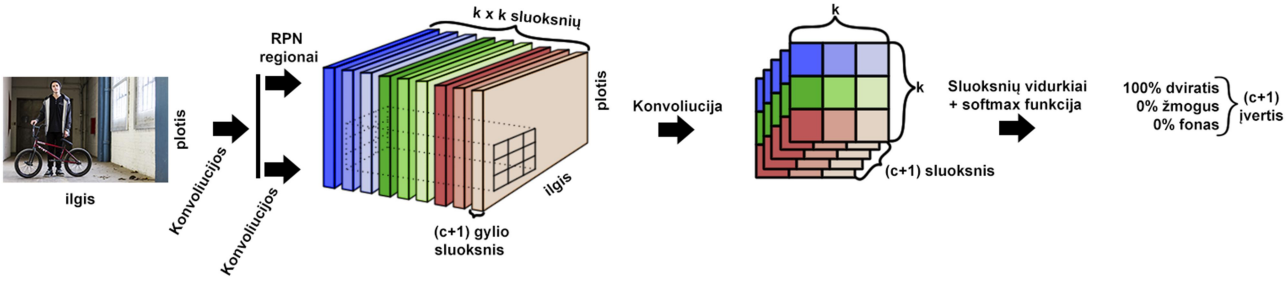


9 pav. Vieno ResNet 101 bloko struktūra.

4.2. Tinklų architektūros

4.2.1. R-FCN

R-FCN architektūros modelis yra dviejų dalių objektus aptinkantis tinklas, kuris savo struktūra yra panašus į Faster R-CNN tinklą. Abu tinklai naudoja tokį pat regionus siūlantį subtinklą, tačiau R-FCN modeliu bandyta išspręsti Faster R-CNN klasifikuojančios modelio dalies problemą. Šiame modelyje atsisakyta skaičiavimų atžvilgiu intensyvių pilnai sujungtų sluoksnių naudotų Faster R-CNN ir jie pakeisti konvoliuciniais sluoksniais. Naujieji konvoliuciniai sluoksniai kiekvienai bandomai aptikti klasei sukuria keletą atskirų bruožų žemėlapių, kur viena plokštuma nusako, kaip tikėtina, kad pasirinktoje srityje yra atitinkamos klasės kažkuri objekto dalis (pav. 10).



10 pav. R-FCN klasifikatoriaus schema.

Iš bruožus formuojančio subtinklo gauti duomenis yra perduodami regionus siūlančiai modelio daliai ir kovoliuciniam klasifikatoriui. Šie keli konvoliuciniai sluoksniai apdoroja gautus duomenis taip, kad gautas išeities gylis taptų *k*k*(c+1)* (kur *k* atitinka į kiek dalių bus sudalintas pasiūlytas regionas vertikaliai bei horizontaliai, o *c+1* reiškia mokomų atpažinti klasių skaičių kartu su fonu). Šioje struktūroje visas klases (su fonu) atitinka po k*k bruožų žemėlapio sluoksnių. Kiekviename jų yra saugomi aktyvacijos svoriai atitinkamos klasės ir regiono dalies. Nustatyta bandymų metu, kad regionus dalinant į $k \times k = 7 \times 7$ dalių, vietoje 3×3, mAP tikslumas pagerėja 1,1% sprendžiant PASCAL VOC 2007 duomenų bazės objektų aptikimą. [DLH+16] R-FCN tinklo klasifikatorius iš RPN gauvęs aibę regionų, kuriuose galimai yra objektai atlieka šiuos veiksmus kiekvienam regionui. Pasirenka regioną atitinkančią dalį iš apdoroto bruožų žemėlapio jį suskirsto į $k \times k$ dalių taip, kad kiekvieną regiono dalį aktyvuojantys sluoksniai būtų sugrupuoti kartu. Kiekvienai šių grupių apskaičiuojami regionų vidurkiai visoms klasėms ir gaunama $k \times k \times (c+1)$ dimensijų struktūra. Šios struktūros kiekvienas elementas reiškia, kaip stipriai tikėtina, kad pasirinkta regiono dalis pateko ant pasirinktos klasės objekto atitinkamos dalies. Šios struktūros kiekvienos plokštumos vidurkių vektorius paduotas softmax (pav. 11) funkcijai grąžina c+1 reikšmę, kuri nurodo tikimybę, kad regionas priklauso kiekvienai iš tyriamų klasių

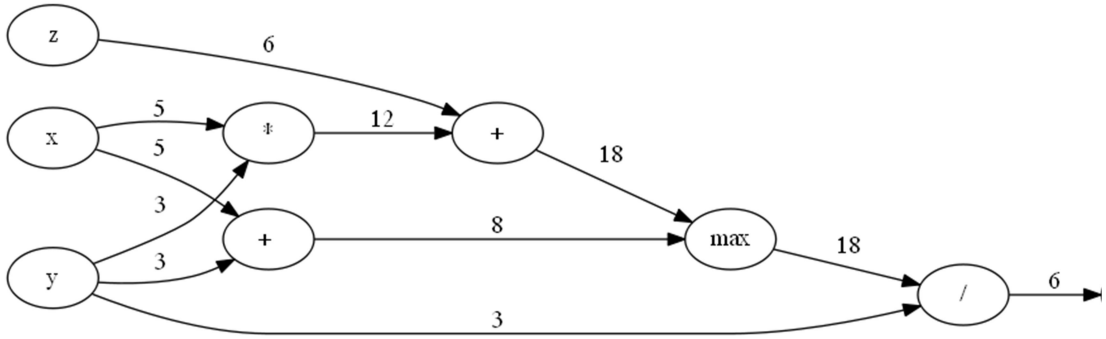
$$f_{softmax}(V) = V' : \{v'_i = \frac{e^{v_i}}{\sum_{i=1}^{|V|} e^{v_i}},\ kur\ v_i \in V\}$$

11 pav. Softmax funkcijos apibrėžimas.

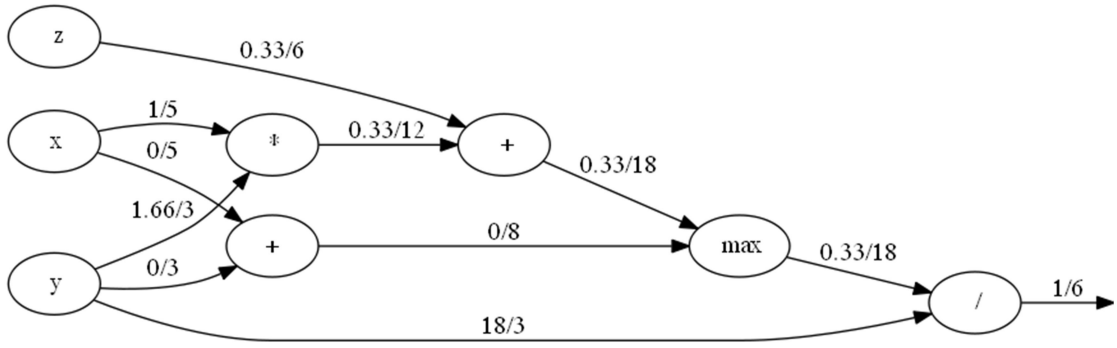
5. Konvoliucinių neuroninių tinklų mokymas

Neuroninių tinklų mokymas yra paremtas optimizavimu. Siekiant išmokyti sistemą teisingai atlikti norimą užduotį yra sprendžiama optimizavimo problema, kurios tikslas sumažinti skirtumą tarp sistemos gauto rezultato ir teisingo rezultato, kurio buvo tikimasi. Objektų atpažinimo uždaviniuose rezultatas, kurio tikimasi, yra nuotraukos etiketėje surašytos regionų koordinatės, dydžiai bei objekto pavaizduoto regione klasės numeris. Skirtumas tarp rezultatų įvertinamas loss funkcija.

Modelio mokyme naudojamas stochastinio nusileidimo gradientu metodas (angl. "Stochastic gradient descent"), kuris pagal dalines išvestines pasirenka, kaip reikia pakeisti tinklo svorius, kad loss funkcijos reikšmė sumažėtų. Šios dalinės išvestinės gaunamos pirma pradinio paveikslėlio duomenis persiunčiant per visą modelį ir su kiekviena operacija įsiminama atėję duomenys ir operacijos rezultatas (pav. 12). Po to naudojant pagal tinklo rezultatą apskaičiuotą loss funkciją yra grįžtama per visas atliktas operacijas ir pakeliui yra surandamos ir išsaugomos dalinės išvestinės pagal kiekvieną sutiktą tinklo svorį (angl. "backpropagation") (pav. 13).



12 pav. funkcijos $f(x,y,z) = \frac{max(x*y+z,\ x+y)}{y}$ duomenų sklidimas gilyn.



13 pav. Grįžimas atgal skaičiuojant gradientus. Kintamųjų gradientai: ∇(z, x, y) = (0,33; 1,00; 19,66)

Šie gradientai nusako, kokiu santykiu (arba optimizavimo žingsnio kryptimi) reikia keisti modelio svorius. Be to, norint gauti naujus svorius pagal stochastinį nusileidimą gradientu naudojamas mokymosi greitis, kuris kokio dydžio naudoti optimizavimo žingsnius (pav. 14).

$$W_{i+1} = W_i - \alpha * \nabla Loss(W_i)$$

14 pav. Naujų svorių skaičiavimo funkcija, kur W yra svoriai, *a* - mokymosi greitis.

Objektų atpažinimo srityje dažnai modelių efektyvumas yra vertinimas naudojant mAP (angl. *"mean average precision"*) tikslumą. Apmokytam neuroniniam tinklui pateikiama nauja mokymo metu nematyta duomenų bazės dalis, pagal kurios modelio gautus rezultatus bus vertinamas objektų aptikimo tikslumas. Naudojant šią naują duomenų dalį vertinimui stengiamasi išvengti mokymo aibės duomenų įsiminimo (angl. "*overfitting"*), kuris atsiranda dėl to, kad mokymo duomenų aibė neapima visų įmanomų tos klasės objektų, todėl tikslumas gali gautis klaidingai didesnis.

PASCAL VOC, vienoje iš didžiausių objektų atpažinimo užduočių, mAP tikslumo skaičiavimas apibrėžtas tokiu būdu [EGW+10]:

1. Visi modelio grąžinti regionai, kurie sutampa IoU santykiu didesniu nei 0,5 (pav. 15) su kuriuo nors nuotraukos etiketėje aprašytu regionu ir yra teisingai klasifikuoti, yra laikomi teisingai atpažintais. Tais atvejais, kai keli regionai sutampa su tuo pačiu nuotraukos etiketėje esančiu regionu, tuomet pirmasis regionas skaitomas, kaip teisingai atpažintas, o visi likę yra klaidingi.

$$IoU = \frac{S_{tinklo\ gautas\ regionas} \cup S_{nuotraukos\ objekto\ regionas}}{S_{tinklo\ gautas\ regionas} \cap S_{nuotraukos\ objekto\ regionas}}$$

15 pav. IoU (sankirtos ir sąjungos santykis, angl. "Intersection over Union")

2. Tikrinant po vieną regioną kartu išsaugomi visų jau patikrintų regionų du santykiai: teisingai atpažintų ir visų atpažintų (angl. "recall") bei teisingai atpažintų ir patikrintų modelio regionų (angl. "precision").
3. Patikrinus visus regionus gaunama santykių porų aibė, iš kurios sudaromi 11 poaibių, pagal recall reikšmes. Pirmame poaibyje visi iš intervalo [0; 1], antrame [0,1; 1], trečiame [0,2; 1] ir taip toliau iki vienuolikto [1; 1]
4. Iš kiekvieno šių vienuolikos intervalų išrenkamos didžiausios precision reikšmės.
5. Visų vienuolikos reikšmių vidurkis yra laikomas apmokyto modelio mAP tikslumu.

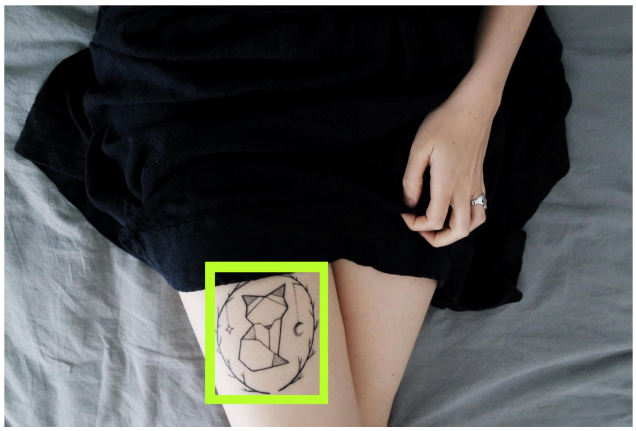
6. Duomenų bazės sudarymas

Neuroniniai tinklai surasti ir atskiri objektus mokosi iš pavyzdžių, todėl norint apmokyti tikslų modelį reikia surinkti didelį kiekį nuotraukų su teisingai sužymėtais ieškomų objektų regionais. Geros objektų atpažinimo duomenų bazės sudarymas yra ilgas ir sudėtingas procesas, nes surinkti duomenys turi atspindėti realios tiriamos objektų aibės bei aplinkos įvairovę. Nuo mokymo duomenų kiekio ir įvairovės priklauso apmokyto modelio rezultatų kokybė ir tikslumas. [CLS+16]

Šiam darbui atlikti buvo naudota duomenų bazė sudaryta iš nuotraukų rinktų Google vaizdų paieškos pagalba. Duomenų bazei naudotos tik JPG formato, trijų kanalų (RGB) nuotraukos su žmonėmis turinčiais tatuiruočių. Pasirinktos nuotraukos, kurios ilgiu arba pločiu viršijo 1280 taškų, buvo suspaustos į šias ribas išlaikant paveikslėlio dimensijų santyki. Iš viso duomenų bazė buvo sudaryta iš 822 nuotraukų, kurios buvo atsitiktine tvarka paskirstytos į dvi grupes: testavimo ir mokymo santykiu 173 / 649. Testavimo duomenų dalis buvo naudojama apmokyto modelio tikslumui įvertinti. Toks duomenų padalijimas užtikrina, kad tikslumas vertinamas su pavyzdžiais, kurių neuroninis tinklas negali būti įsiminęs.

Vien nuotraukų neužtenka modeliui išmokti aptikti objektus. Į duomenų bazę kartu su kiekvienu paveikslėliu įeina po etiketę. Ją sudaro nuotraukos dimensijos ir sąrašas objektų regionų, iš kurių kiekvieną sudaro keturi skaičiai (viršutinio kairiojo kampo koordinatės ir regiono ilgis su pločiu) (pav. 16). Sudarant tokių regionų tatuiruočių duomenų bazę atsirado problema, nes tatuiruotės, skirtingai nei kitų dažnai nagrinėjamų sričių objektai (pvz.: žmonės, gyvūnai, mašinos), neturi visad aiškias ribas. Ištatuiruotose frazėse dažnai randami aiškūs tarpai tarp žodžių, tačiau vis vien galima tai laikyti kaip vientysą tatuiruotę. Iš kitos pusės,

kartais kelias senas tatuiruotes žmonės apjungia naujais raštais ar gradientais. Siekiant išvengti tokių dviprasmiškumų, sudarant tatuiruočių aibę, buvo vadovaujamasi tokia tvarka: dvi tatuiruotės buvo žymimos kaip atskiros tuomet, kai skyrėsi jų stiliai, neturėjo jos aškių tiesioginių jungčių arba buvo pavaizduotos viena prie kitos, bet tik dėl žmogaus pozos nuotraukoje (pav. 17).



16 pav. Tatuiruotė neišlenda už regiono ribų, o jas tik liečia.



17 pav. Tatuiruotės liečiasi, bet yra ant dviejų rankų, todėl regionai yra atskiri.

Dumenų bazėje per visas nuotraukas iš viso buvo sužymėti 1880 regionų (iš kurių 1558 mokymo dalies nuotraukose ir 322 testavimo dalyje).

7. Duomenų bazės modifikavimas

Didelės duomenų bazės leidžia modeliui geriau išmokti bendresnius objektų požymius ir bruožus. Nuo duomenų kiekio ir įvairovės priklauso bet kokio modelio tikslumas, tačiau ne visados įmanoma duomenų baze apimti visą objektų ir situacijų įvairovę arba duomenų rinkimas ir apdorojimas yra per daug sudėtingas ar brangus laiko ir pinigų atžvilgiu. Vienas iš paprastų ir gan efektyvių būdų gauti daugiau mokymo duomenų yra turimą duomenų bazę modifikuoti. Be to, primityvūs pokyčiai duomenų aibėje padeda modeliui išmokti ignoruoti nereikšmingą informaciją ar triukšmą. Kita problema, kurią duomenų modifikavimas padeda išspręsti yra tiriamos srities neatitikimas. Tais atvejai, kai duomenų bazėje tiriami objektai arba jų aplinka nėra tokia įvairi, kaip realiame pasaulyje, galima modifikavimais imituoti trūkstamą įvairovę (pvz. jei bandant aptikti automobilius trūksta pavyzdžių tamsiu paros metu, tuomet galima nuotraukas darytas diena patamsinti). Tačiau ne visoms duomenų aibėms visos transformacijos yra prasmingos ar padedančios. Nuotraukų apsukimas, mokant tinklą aptikti automobilius, padvigubina aibę tinkamais duomenimis, bet bandant atpažinti skaičius ar raides, tokia pat transformacija nėra teisinga, nes kai kurių simbolių simetriškos poros yra kiti simboliai.

Duomenų bazės modifikavimo procesus galima padalinti į tokias grupes pagal jų sudėtingumą ir atlikimo laiką:

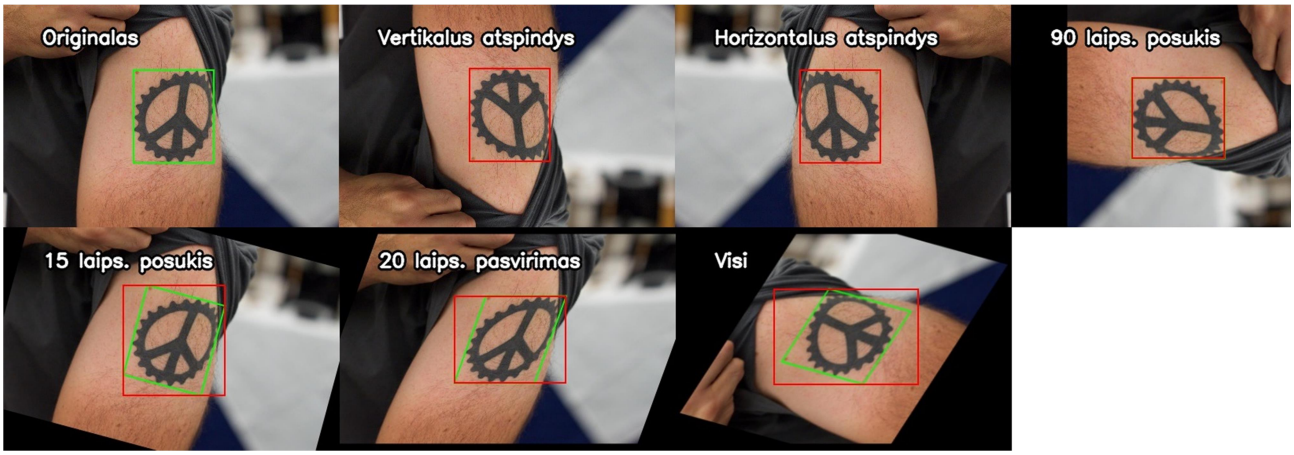
- Išankstinis duomenų modifikavimas (angl. *"offline augmentation"*) - pavyzdžiai yra sukuriami prieš pradedant mokyti tinklą ir išsaugomi diske. Šiuo būdu sukuriamų pavyzdžių nebūna daug, nes dažniausiai tai atliekama, kai pačios modifikacijos yra sudėtingos. Kita priežastis, kodėl gali būti generuojami iš anksto pavyzdžiai yra ta, kad ne visi sukurti duomenys yra tinkami mokymui ir prieš naudojant juos gali prireikti žmogaus peržiūros ir blogų pavyzdžių išmetimo.
- Modifikavimas modeliui "prireikus" (angl. *"augmentation on the fly"*) - mokymo metu, kiekvienas naujas pavyzdys prieš patekdamas į modelį yra kas kartą vis kitaip

apdorojamas. Pačios modifikacijos nėra sudėtingos, todėl jos dažnai atliekamos kompiuterio procesoriaus, kol grafinė plokštė dirba su modeliu ir prieš tai paduotu paveikslėliu. Dėl šios priežasties jei modelis pakankamai didelis, o duomenų modifikacijos greitos, toks nuotraukų keitimas beveik neprailgina modelio mokymo laiko.

- Mišrus duomenų modifikavimas - toks mokymo procesas, kai dalis sudėtingų duomenų modifikacijų yra atlikta prieš mokymą, o mokant modifikuojami duomenys papildomai greitesnėmis augmentacijomis.

7.1. Afininės transformacijos

Objektai nuotraukose gali būti pavaizduoti iš skirtingų kampų ar orientacijų. Norint išmokyti tinklą geriau ignoruoti objektų orientaciją aptikimo metu reikia pritaikyti duomenų bazę, kad mokymo pavyzdžiai būtų įvairesni. Afininės transformacijos leidžia imituoti turimų objektų pavyzdžių krypties ar padėties kitimą aplinkoje. Šios transformacijos ypač dažnai sutinkamos kompiuterių regos užduotyse, nes daugumoje jų objektų orientacija ir padėtis nėra svarbi informaciją, pagal kurią sprendžiama pasirinkta užduotis. Nors patiems paveikslėliams afininės transformacijos sukuria naujų pavyzdžių, kuriuose žmogus sugeba atpažinti objektus, tačiau ne visada papildoma informacija reikalinga mokymui yra taip pat atspari šiems pokyčiams. Nagrinėjant objektų atpažinimą naudojant stačiakampius regionus, tokios transformacijos, kaip ne statūs posūkiai, pasvyrimai ir perspektyvos keitimas mokymo duomenims sukuria nereikalingas papildomas paklaidas. Su tokiomis transformacijomis naujos stačiakampės ribos surandamos iš pradinių keturių taškų, todėl jos po kiekvieno modifikavimo apima vis didesnį regioną. Kitose kompiuterių regos užduotyse, kaip klasifikavime, kur jokie regionai nenaudojami, arba semantiniame segmentavime, kur regionai sužįmėti nuotraukos pikselio tikslumu, po tokių transformacijų naujų pavyzdžių aprašymų koregavimas yra nereikalingas arba trivialus. Naudojant tokias transformacijas, kaip 90 laipsnių posūkiai, dydžio keitimai ar atspindžiai, pažymėti stačiakampai regionai išlaiko savo formą ir išlaiko mokymo duomenų tikslumą, todėl tokie modifikavimai dažniau sutinkami objektų aptikimo modeliuose.



18 pav. Afininių transformacijų pavyzdžiai. Žalia spalva pažymėtas tatuiruotės ribos prieš transformacijas, raudona - po transformacijų.

7.2. Kelių pavyzdžių suliejimas

Dviejų pavyzdžių atsitiktinis suliejimas pirmiausiai buvo pradėtas naudoti klasifikavimo užduotyse. [ZCD+17] Šis duomenų modifikavimo būdas klasifikavimo srityje naudotas siekiant duomenims sukurti natūraliai atrodančio papildomo triukšmo. Tačiau objektų atpažinimo srityje parinkus kitokį atsitintinio suliejimo pasiskirstymą sugeneruoti vaizdai pasidaro panašūs į vaizdų suliejima mažo kadrų dažnio kameros. Tokiu būdu sukurtose nuotraukose, priklausomai nuo suliejimo koeficiento kartais matosi abiejų pradinių nuotraukų objektai. [ZHZ+19] Tad šis metodas ne tik sukuria papildomo triukšmo, kurį neuroninis tinklas bando išmokti ignoruoti, bet ir padaugina viename paveikslėlyje ieškomų objektų skaičių.

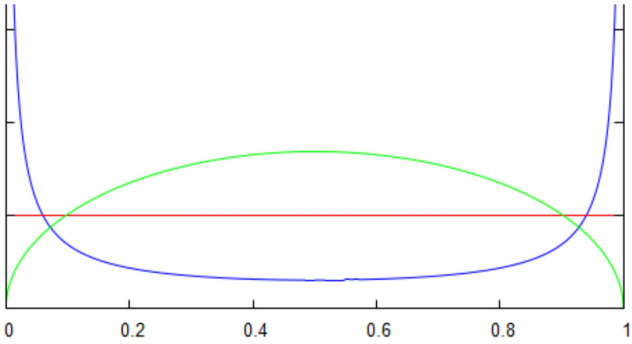
Toks dviejų nuotraukų suliejimas atliekamas sekančiu būdu:

1. Iš duomenų bazės atsitiktinai pasirenkamos dvi nuotraukos, kurios bus suliejamos.
2. Suvienodinami nuotraukų dydžiai vienu iš dviejų būdų: abi nuotraukos praplečiamos iki jų ilgio ir pločio maksimumo, trukstančius pikselius užpildant nuliais (juoda spalva), arba nuotraukos yra ištempiamos / suspaudžiamos iki jų ilgių ir pločių vidurkio (pav. 19).
3. Pataisomi abiejų nuotraukų objektų regionai, kad koordinatės atspindėtų naujas paveikslėlio dimensijas ir sujungiami į vientysą sąrašą.
4. Pagal pasirinktą distribuciją sugeneruojamas atsitiktinis suliejimo koeficientas, kuris nusako, kokia ryškumo dalis bus naudojama pirmajai nuotraukai.
5. Sukuriamas regionų svorių sąrašas, kur kiekvienam regionui iš pirmos nuotraukos priskiriamas pasirinktas suliejimo koeficientas, o regionams iš antrosios - iš vieno atimtas pasirinktas koeficientas.
6. Visi pirmos praplėstos nuotraukos taškai yra padauginami iš pasirinkto suliejimo koeficiento, o atrosios nuotraukos sudauginami su iš vieno atimto koeficiento. Po sandaugos abiejų nuotraukų atitinkami elementai sudedami.



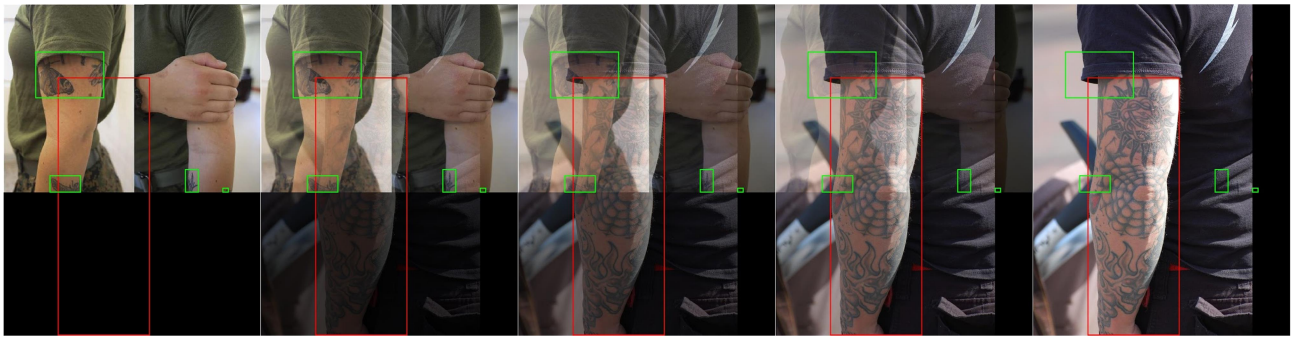
19 pav. Pirmos dvi nuotraukos yra pradinės nuotraukos, trečia buvo gauta pirmas dvi nuotraukas praplėtus iki vienodų dimensijų, o ketvirtoje - abi nuotraukos ištemptos, kad būtų vienodų formų.

Nuo suliejimo koeficiento priklauso, kuri iš nuotraukų bus ryškesnė. Siekiant dar labiau paįvairinti turimą duomenų aibę, šis koeficientas nėra pastovus, o pasirenkamas atsitiktinai naudojant beta distribucija. Ši distribucija priklauso nuo dviejų kintamųjų. Jeigu jie abu yra mažesni už 1, tuomet distribucija stipresnė intervalo [0;1] kraštuose, jei abu didesni už 1, tuomet stipresnė viduryje. Naudojant skirtingus koeficientus ši distribucija tampa nesimetriška. Klasifikavimo srityje naudojami kintamieji mažesni už 1 siekiant sukurti papildomo triukšmo, o objektų aptikimo uždaviniuose bandymų metu aptikta, kad naudojant abu kintamuosius lygius 1.5 yra pasiekiamas didžiausias modelio tikslumas.



20 pav. Beta pasiskirstymai su skirtingai koeficientais. Raudona spalva B(1, 1), mėlyna B(0.2, 0.2), žalia B(1.5, 1.5).

Iš suliejimo koeficientų sukurtas regionų svorių sąrašas naudojamas skaičiuojant loss funkciją. Šie svoriai atitinka kiekvieno regiono įtaką skaičiuojant paklaidas, tam, kad regionai iš mažiau ryškios nuotraukos, kuriuos sunkiau aptikti, nesugadintų gero ryškesnios dalies rezultato (pav. 21).



21 pav. Kaip nuotraukų suliejimo koeficientas keičia modifikacijos rezultatą. Naudoti suliejimo koeficientai atitinkamai 0, 0.25, 0.5, 0.75 ir 1.

8. Neuroninio tinklo žinių perdavimas

Gilūs tinklai pasižymi pranašumu sprendžiant įvairias kompiuterių regos užduotis. Nors gilių architektūrų modeliai sugeba pasiekti tikslesnius rezultatus, tačiau už šį pranašumą tenka sumokėti laiko sąskaita. Žinių perdavimas iš tikslaus, bet sudėtingo, modelio paprastesniam leidžia išlaikyti mažesnio tinklo greitį ir priartinti tikslumą giliam tinklui. Norint pasiekti šiuos rezultatus pirma yra apmokomas būsimas mokytojas, gilus neuroninis tinklas, ir sustabdomi visi jo kintamieji, kad mažojo tinklo - mokinio - mokyme nesikeistų. Paprastesnio tinklo mokymo metu kiekvienas pavyzdys iš mokymo duomenų aibės yra paduodamas abiems tinklams ir vietoje įprasto klaidų įverčio (tarp rezultato ir mokymo duomenų) yra įvedamas papildomas skirtumas tarp gilaus ir paprastojo tinklo duomenų.

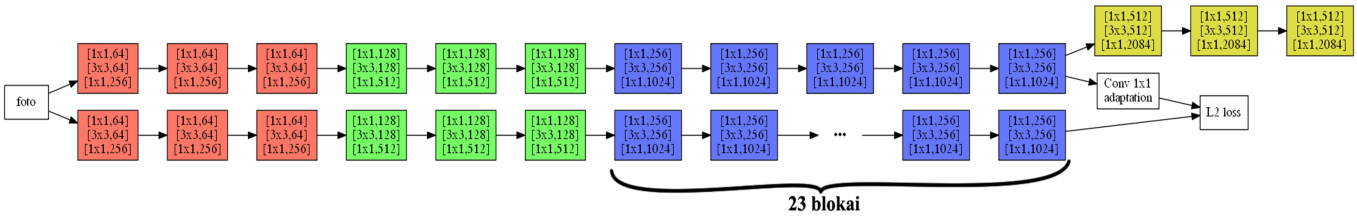
Dviejų dalių objektų aptikimo tinkluose naudojant žinių perdavimą yra pakeičiamos regionus siūlančio subtinklo ir regionus klasifikuojančios dalies skirtumo (loss) funkcijos prijungiant papildomą paklaidą nuo gilaus tinklo. Be šių pakoreguotų paklaidų kartu sukuriama nauja jungtis tarp abiejų tinklų ir įvedama bruožų žemėlapių loss funkcija. Ji analogo įprastame mokyme neturi ir jos veikimas detaliau nagrinėjamas šiame darbe. Jungties tarp modelių pagrindinė paskirtis yra iš tinklo mokytojo suteikti užuominas mokiniui. Bruožų žemėlapiuose sutelkiama visa reikiama informacija tolimesniam darbui su regionais, todėl pagerinus jų kokybę kartu pagerėja ir viso modelio tikslumas. Iš abiejų modelių yra pasirenkama viena arba kelios poros atitinkamų sluoksnių, kurių rezultatus bandoma suvienodinti. Kiekvienoje poroje prie mokinio sluoksnio yra prijungiamas adaptacijos 1x1

konvoliucijos sluoksnis, kuris padeda rezultatą pritaikyti mokytojo gautam. Šis sluoksnis suvienodina bruožų žemėlapių formą, kad būtų galima apskaičiuoti skirtumą tarp jų. Tačiau net ir tais atvejais, kai pasirinktoje mokytojo ir mokinio poroje sutampa bruožų žemėlapio forma, šis adaptacijos sluoksnis padeda pagerinti mokinio rezultatą. Skirtumas tarp sluoksnių skaičiuojamas naudojant L2 skirtumo funkciją, kuri gauna kiekvienos elementų poros skirtumų kvadratų sumą. Prieš prijungiant šį skirtumą prie likusio modelio loss funkcijos jis

$$L_2loss = \sum_{i=1}^{n}(m_{1i} - m_{2i})^2$$

yra dauginamas su užuominų svorio, kuris nurodo šio skirtumo įtaką mokymui.

Šiame darbe naudojant du R-FCN tinklus su skirtingo gylio ResNet bruožus formuojančiais subtinklais buvo bandomas toks žinių perdavimas užuominomis. Tinklas su ResNet 101 bruožus formuojančiu subtinklu atstovavo mokytoją, o ResNet 50 turintis tinklas buvo mokinys. Mokytojas su mokiniu sujungtas po paskutinio trečio tipo blokų (pav. 22).



22 pav. Supaprastintos ResNet 50 (viršuje) mokinio ir ResNet 101 (apačioje) mokytojo architektūros. Kiekvienas blokas sudarytas iš trijų konvoliucijų, vienodi blokai nuspalvinti ta pačia spalva.

9. Bandymai

Visi šiame darbe naudoti tinklai buvo paremti R-FCN architektūra. Tinklai buvo mokomi aptikti tatuiruotes nuotraukose naudojant surinktą duomenų bazę. Du iš šių modelių buvo mokomi naudojant mokymo pavyzdžių suliejimą:

- Pavyzdžiai praplečiami užpildant papildomą plotą juodai.
- Pavyzdžiai ištempiami, kad sutaptų jų dydžiai

Abu tinklai naudojo ResNet 101 bruožus formuojantį subtinklą. Palyginimui buvo apmokytas R-FCN ResNet 101 tinklas, kuris buvo identiškas dviems pirmiems tinklams, tačiau jo mokyme nenaudotas pavyzdžių suliejimas.

Be to, šis įprastai apmokytas modelis buvo naudojamas kaip mokytojas tiriant, kaip priklauso tinklo kokybė nuo mokytojo užuominų svorio. Mokiniai tinklai naudojo bruožus formuojantį ResNet 50 subtinklą, kuris turi per pusę mažiau sluoksnių nei mokytojas. Bandymai atlikti naudojant 0.5, $1*10^{-5}$, $1*10^{-6}$, $1*10^{-7}$, $1*10^{-8}$ užuominų svorius bei palyginimui apmokytas toks pat tinklas be mokytojo užuominų.

Pastaba: tiek mokant modelius naudojant pavyzdžių suliejimą, tiek mokantis su mokytojo užuominomis kartu buvo naudojamas horizontalus pavyzdžių atspindėjimas.

Neuroninio tinklo mokymasis ir galutinis tikslumas priklauso nuo įvairių mokymo ir architektūros parametrų, kurie pasirenkami prieš pradedant mokymą ir nėra optimizuojami mokymo proceso. Šie parametrai visoms architektūroms buvo naudoti tokie patys:

- Mokymosi greitis (angl. “learning rate”) nurodo kokio dydžio žingsnis yra daromas mokymosi metu, gradientui priešinga kryptimi (gradientas nurodo kryptį, kuria

skirtumas didėja, todėl pasirenkama gradientui atvirkčia kryptis). Naudojant nedidelį mokymosi greitį neuroninis tinklas mokosi daug lėčiau. Iš kitos pusės, naudojant didesnį greitį, lokalų minimumą galima surasti greičiau, tačiau jis dažnai būna blogesnis, arba dėl per žingsnio dydžio mokosi nestabiliai, kol rezultatas pradeda visai blogėti ir apmokyti nepavyksta. Renkantis tinkamą greitį reikia atrasti pusiausvyrą tarp mokymosi trukmės ir tikslumo. Be to, šį kintamąjį galima keisti mokymo metu, tai yra pradžioje naudoti didesnį, kol tinklui yra toli iki gero užduoties rezultato, o vėliau po truputį jį mažinti, kad geriau priderinti ir gauti tikslesnį modelį. Mokant visus modelius buvo naudotas 0,0003 mokymo greitis.

- Iteracijos grupės dydis (angl. “batch size”) - šis parametras nurodo, kiek nuotraukų bus naudojama vienu metu, kai skaičiuojamas gradientus. Apdorojus visas grupės nuotraukas yra surandamas gradientas ir tada daromas vienas optimizavimo žingsnį. Didesnis grupės dydis padeda tinklui atlikti optimizavimo žingsnį pagal bendresnę informaciją. Nuo grupės dydžio priklauso reikalingos mokymui kompiuterio atminties kiekis. Neuroninių tinklų mokymui buvo pasirinktas maksimalus grupės dydis, su kuriuo nepritrūko kompiuterio atminties (naudotas grupės dydis R-FCN tinkle buvo 1).

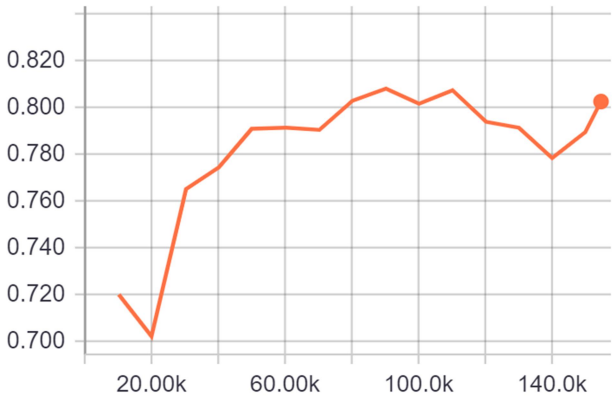
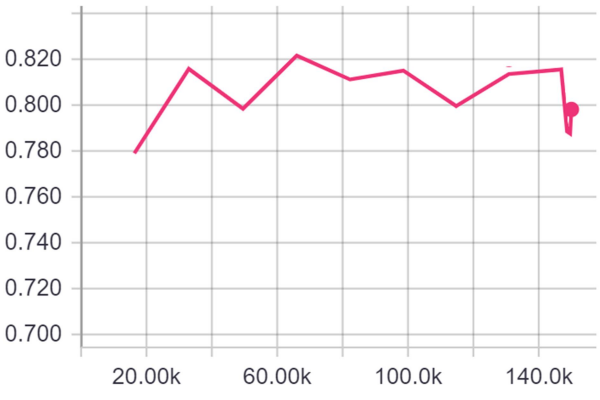
Visi modeliai buvo apmokomi ir testuojami naudojant nešiojamą kompiuterį su Intel Core i7-4720HQ procesoriumi, NVIDIA GeForce GTX 960M vaizdo plokšte. Modeliai buvo mokomi iki 150 000 žingsnių ir įvertinti keliolika kartų mokymo metu įvairiose tarpinėse būsenose (angl. “checkpoints”).

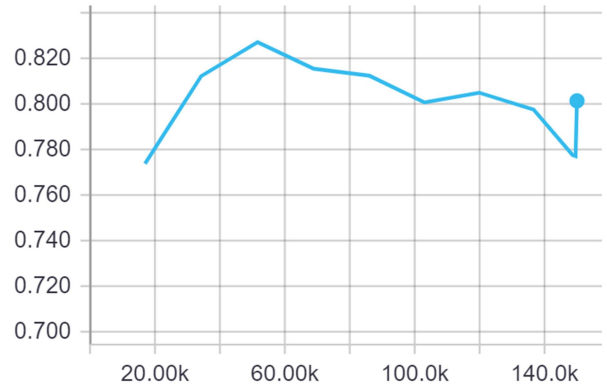
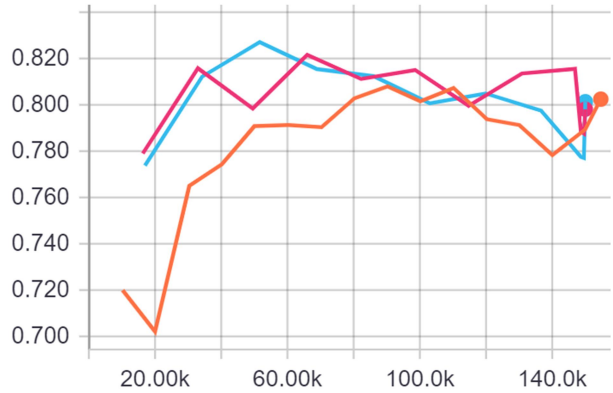
9.1. Pavyzdžių suliejimo rezultatai

Modelio mokymo būdas	**Geriausiai pasirodžiusios tarpinės būsenos mAP tikslumas**
Įprastas	0,8079
Pavyzdžiai sulieti praplečiant paveikslėlį	0,8212
Pavyzdžiai sulieti ištempiant paveikslėlį	0,8271

23 pav. Bandymo rezultatai.

Tikslumas mAP buvo vertinamas maždaug kas 15 000 žingsnių (pav. 23)

Įprastas	Pavyzdžiai sulieti praplečiant paveikslėlį
	

Pavyzdžiai sulieti ištempiant paveikslėlį	Visi
	

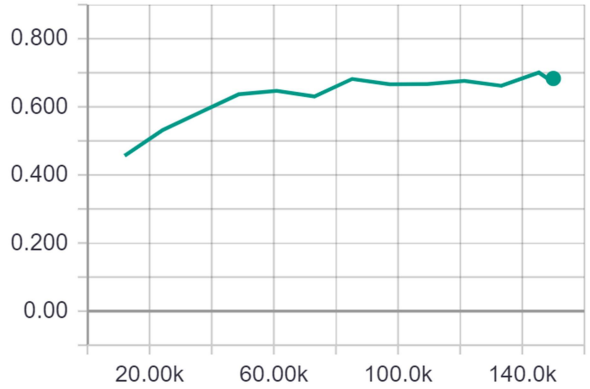
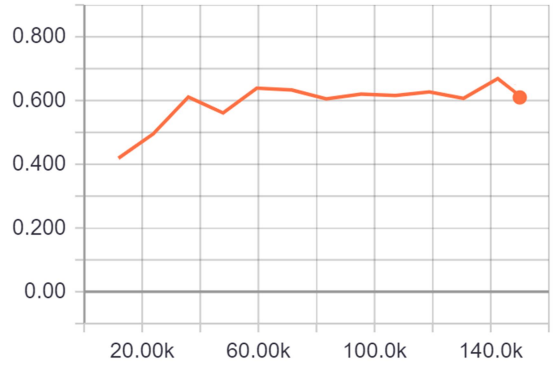
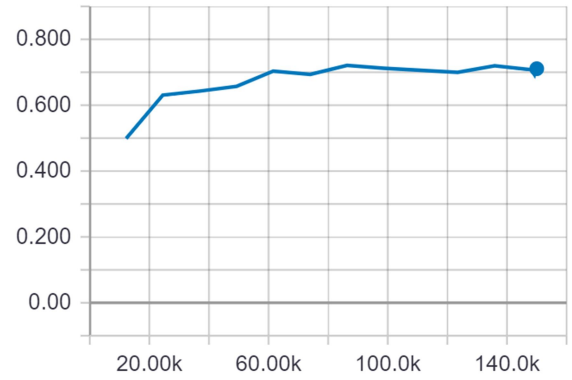
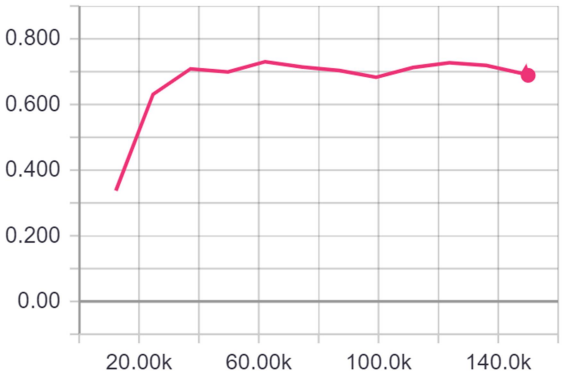
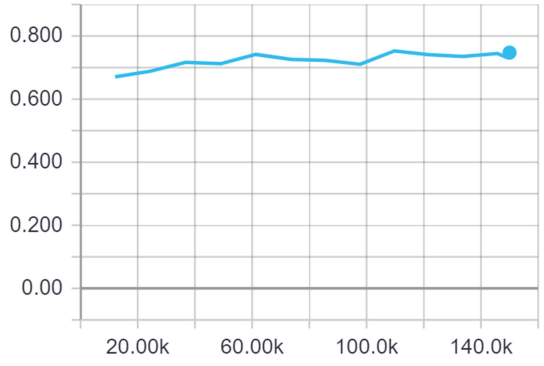
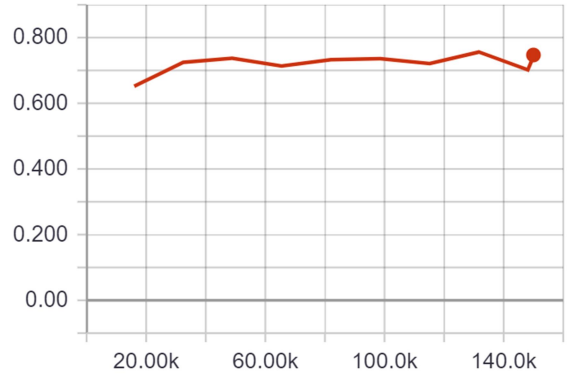
24 pav. Kiekvieno modelio tikslumas mokymo metu (vertikaliai žymimas tikslumas, horizontaliai - žingsnių skaičius). Oranžine spalva pavaizduotas įprastai mokytas modelis; rožinė - su pavyzdžiais sulietais praplečiant paveikslėlį; mėlyna - su pavyzdžiais sulietais ištempiant paveikslėlį.

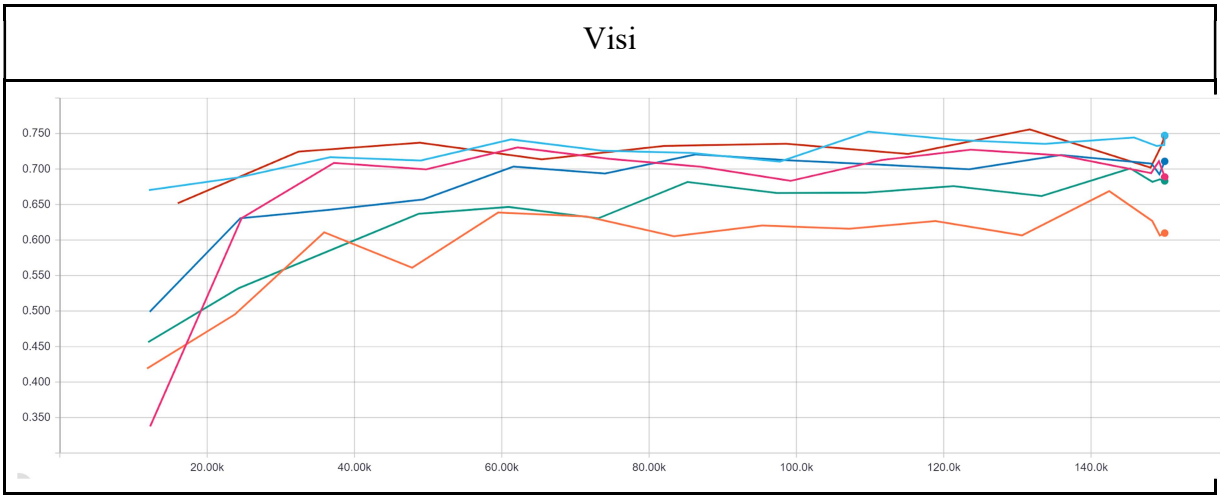
Tais atvejais, kai paveikslėlių dimensijos nesutampa, praplečiant nuotraukas ir trūkstamą plotą užpildant nuliais ne visa erdvė yra išnaudojama optimaliai. Tokiu būdu suliejus visas naudotos tatuiruočių duomenų bazės nuotraukų kombinacijas gautas paveikslėlis vidutiniškai buvo sudarytas iš 48% nuotraukų sankirtos, 41% ploto sukurto tik iš vienos nuotraukos, 11% nuliais užpildyto ploto. Šią problema buvo bandoma išspręsti abu paveikslėlius ištempiant / suspaudžiant iki formos vidurkio, kad abu pavyzdžiai gautųsi kuo mažiau pasikeitę. Tačiau šita intuicija, kad pirmasis suliejimo būdas yra mažiau efektyvus, nei paveikslėlių ištempimas, nepasitvirtino. Nors abiejų modelių rezultatai triukšmingi, tačiau vertinant tikslumus pasirodė apylygiai ir aplenkė ~2% įprastu būdu apmokytą modelį. Gali būti, kad didžioji dalis šio duomenų modifikavimo būdo efektyvumo slypi tame, kad kiekvienoje nuotraukoje atsiranda daugiau objektų pavyzdžių, o papildomas suliejimo triukšmas ne tiek reikšmingas antrinis tokio suliejimo produktas.

9.2. Žinių perdavimo rezultatai

Užuominų mokymo svoris	**Geriausiai pasirodžiusios tarpinės būsenos mAP tikslumas**
0.5	0,7008
0.00001	0,6689
0.000001	0,7206
0.0000001	0,7304
0.00000001	0,7526
Įprastai apmokytas modelis	0,7558

25 pav. Bandymų rezultatai
Tikslumo grafikai, vertinama kas ~15 000 žingsnių

0.5 užuominų svoris	0.00001užuominų svoris
	
0.000001 užuominų svoris	**0.0000001 užuominų svoris**
	
0.00000001 užuominų svoris	**Įprastai apmokytas modelis**
	



26 pav. Modelio tikslumo kaita mokymo metu (vertikaliai žymimas tikslumas, horizontaliai - žingsnių skaičius). Žalia spalva pavaizduotas modelis mokytas su 0.5 užuominų svoriu; oranžinė - 10^{-5}; mėlyna - 10^{-6}; rožinė - 10^{-7}; šviesiai mėlyna - 10^{-8}; raudona - įprastai mokytas modelis.

Mokant modelį naudojant žinių perdavimą su rekomenduojamą 0,5 užuominų skirtumo svoriu buvo pasiektas beveik blogiausias tikslumo rezultatas. Šiuo būdu mokant modelį užuominų sukuriama skirtumo dalis užgožia regionų siūlimą ir klasifikavimą. Mažinant užuominų skirtumo svorį rezultatas gerėja, tačiau jis tik artėja prie įprasto mokymo gauto tikslumo, bet jo neaplenkia. Gali būti, kad tokiam žinių perdavimui geriau pasireikšti reikia didesnio skirtumo tarp mokytojo ir mokinio tinklų rezultato arba nepakanka vien standartinių klasifikavimo ir regionų skirtumų funkcijų.

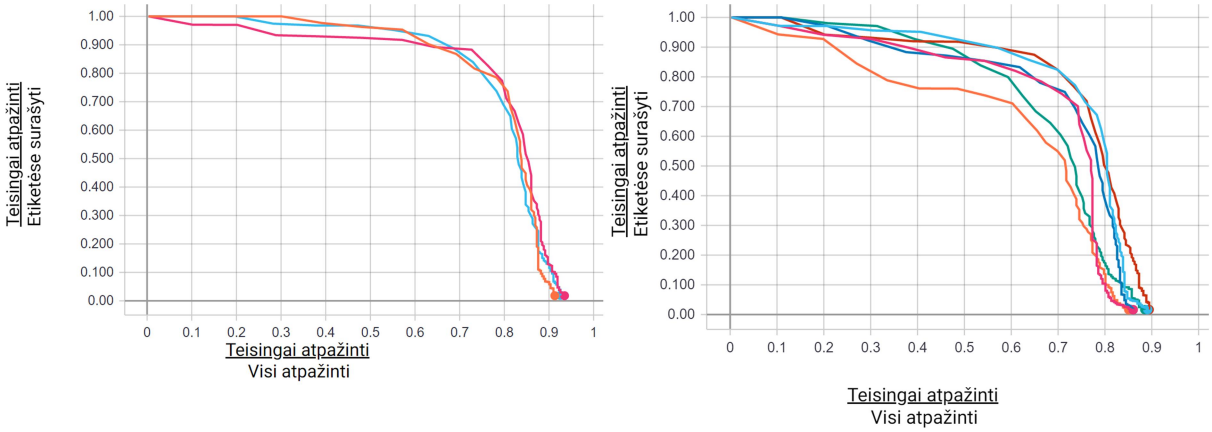
9.3. PR kreivės

PR (angl. “Precision-recall”) kreivės padeda įvertinti modelio rezultatus lyginant teisingai atpažintų (angl. “true positive”), neteisingai atpažintų (angl. “false positive”) bei neteisingai neatpažintų (angl. “false negative”) regionų kiekių santykius. Šio tipo grafikuose vertikalioje ašyje žymimas santykis tarp teisingai modelio atpažintų ir nuotraukų etiketėse surašytų realių regionų, o horizontalioje ašyje santykis tarp teisingai atpažintų ir visų modelio atpažintų regionų.

- Teisingai atpažintais (true positive) regionais laikomi tie, kurie sutampa su nuotraukos etiketėje pažymėtu regionu (IoU > 0,5 su vienu etiketės regionu).
- Teisingai neatpažintų (true negative) regionų nėra, nes etiketėse yra surašyti tik tie regionai, kuriuos turi atpažinti teisingai, ir tinklas pateikia tik regionus, kuriuose mano, kad yra objektas.
- Neteisingai atpažinti (false positive) yra tie, kuriuos tinklas tinklas atpažino, bet nuotraukos etiketėje jo nėra (IoU < 0,5 su kiekvienu etiketės regionu) arba su etiketėje esančiu regionu jau kirtosi kitas tinklo aptiktas regionas.
- Neteisingai neatpažinti (false negative) regionai tokie, kurie yra aprašyti nuotraukos etiketėje, bet nei vienas tinklo atpažintas regionas nesutampa su juo.

Grafo taškai yra apskaičiuojami iš eilės vertinant visus turimus pavyzdžius ir po kiekvieno įsiminant tarpines “precision” ir “recall” reikšmes.

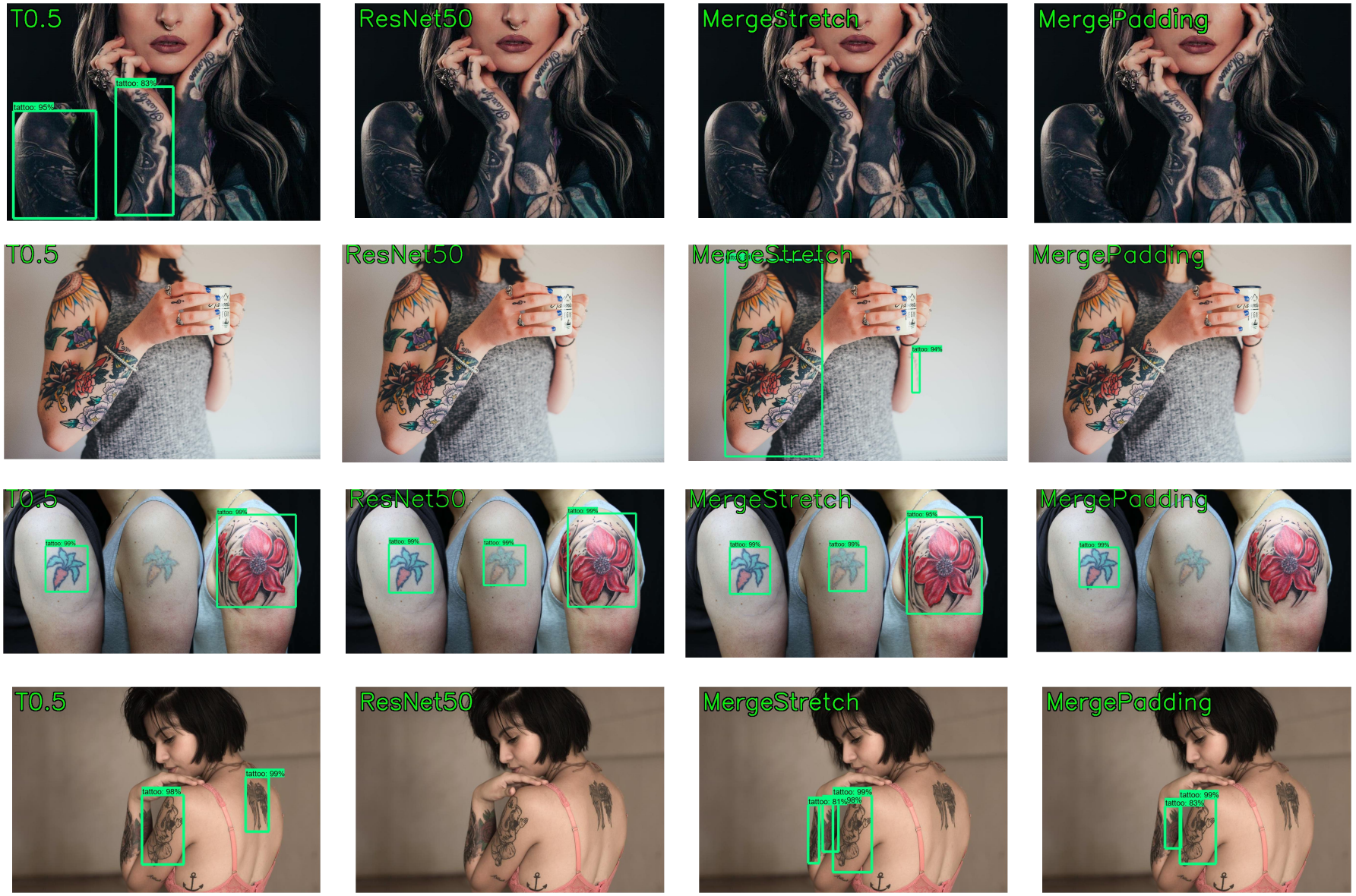
Visų modelių PR kreivės (pav. 27)



27 pav. Kairėje pusėje pavyzdžių suliejime naudotų modelių PR kreivė. Oranžine spalva pavaizduotas įprastai mokytas modelis; rožinė - su pavyzdžiais sulietais praplečiant paveikslėlį; mėlyna - su pavyzdžiais sulietais ištempiant paveikslėlį. Dešinėje pusėje - modeliai mokyti su žinių perdavimu. Žalia spalva pavaizduotas modelis mokytas su 0.5 užuominų svoriu; oranžinė - 10^{-5}; mėlyna - 10^{-6}; rožinė - 10^{-7}; šviesiai mėlyna - 10^{-8}; raudona - įprastai mokytas modelis.

10. Pavyzdžiai




T0.5
ResNet50
MergeStretch
MergePadding
tattoo: 95%
tattoo: 83%
tattoo: 94%
tattoo: 99%
tattoo: 98%
tattoo: 81%



28 pav. Kiekvieno apmokyto modelio tatuiruočių aptikimo pavyzdžiai, iš viršaus žemyn: duomenų bazės etiketėse pažymėti regionai; įprastai apmokytas tinklas su ResNet-101 subtinklu; modelis apmokytas suliejant pavyzdžius praplečiant nuliais; apmokytas suliejant pavyzdžius ištempiant; įprastai apmokytas tinklas su ResNet-50 subtinklu; žinias perduodant su 0,5 svoriu; su 10^{-6} svoriu; su 10^{-8} svoriu. Nuotraukos atrinktos iš duomenų bazės, paskutinė - generuota ir apkarpyta.

11. Išvados

Sistema, kuri sugebėtų teisingai atpažinti žmones, turi veikti efektyviai, kad gavusi didelį kiekį pavyzdžių greitai juos apdorotų. Šį procesą galima paspartinti iš nuotraukų atrinkus tik svarbią identifikavimui informaciją. Tatuiruotės pasižymi unikalumu bei ilgalaikiškumu, dėl šios priežasties jos puikiai tinka tokiam žmonių identifikavimui. Pasitelkus naujas technologijas, kaip konvoliucinius neuroninius tinklus skirtus vaizdų aptikimui, tatuiruočių suradima paveikslėliuose galima pagreitinti ir automatizuoti.

Šio darbo eigoje apmokyti keli modeliai naudojant žinių perdavimą siekiant sukurti tikslią ir greitai veikiantį modelį. Mokymas su papildomai pridėtomis užuominomis pasirodė neefektyvus. Keičiant užuominų įtaką mokymui rezultatai gaunami blogesni arba panašūs į įprastu būdu apmokyto modelio. Dėl šios priežasties reikėtų mokytojo ir mokinio jungtis sudaryti kitoje bruožus formuojančio subtinklo vietoje ir nenaudoti įprastų klasifikavimo ir regionų skirtumų skaičiuojančių funkcijų.

Be to, šiame darbe naudojant pavyzdžių suliejimo modifikacijas gauti rezultatai triukšmingi, tačiau matomas nedidelis modelio tikslumo pagerinimas. Kartu buvo pastebėta, kad suvienodinant nuotraukų dydžius paruošiant suliejimui, nėra didelio skirtumo ar paveikslėliai praplečiami nuliais ar ištempiami iki reikiamos formos. Nagrinėjant šią temą toliau galima būtų ištirti, ar objektų pavyzdžių kiekio padidinimas vienoje nuotraukoje yra svarbioji šios modifikacijos dalis ar suliejimo sukuriamas triukšmas irgi prisideda prie didesnio tikslumo.

12. Summary

Tattoos being unique and everlasting allow them to be used for people recognition. If there was a system that was capable of doing this task it would have to be able to find tattoos in a picture and compare them to other pictures. This paper only focuses on the first part of the process. With database of examples and training convolutional neural networks are able to find and recognise various objects in images. Using a pre-trained neural network helps to cut down on the time and data needed to achieve good object detection results. In this paper I try out two ways of improving performance of a network. The first one is data augmentation method that merges pictures to achieve better results. And the second is knowledge transfer using hints in hopes of improving the accuracy of a smaller network using a better but slower teacher neural network. Using an R-FCN ResNet-101 object detection network I was able to achieve 0.821 (+0.014 improvement over baseline) and 0.827 (+0.02) mAP score using picture merge with padding and stretching respectively used to match image sizes. For knowledge transfer baseline R-FCN ResNet-101 model was used as a teacher and several R-FCN ResNet-50 models as students. Teacher and student models were connected with one link in the feature extractor (before the last 3 conv. layers). Student used 1x1 conv. adaptation layer and L2 loss for the hints. After trying with several hint weights (0.5, $1*10^{-5}$, $1*10^{-6}$, $1*10^{-7}$, $1*10^{-8}$) none of the models improved over ResNet-50 baseline (0,7558 mAP) and using lower weight only converged to the baseline. All the models were trained using labeled tattoo images collected using Google image search.

Literatūros sąrašas:

[BPL10] YL. Boureau, J. Ponce, Y. LeCun, A Theoretical Analysis of Feature Pooling in Visual Recognition, URL http://yann.lecun.org/exdb/publis/pdf/boureau-icml-10.pdf 211KB , 2010

[CLS+16] J. Cho, K. Lee, E. Shin, G. Choy, S. Do. How much data is needed to train a medical image deep learning system to achieve necessary high accuracy? URL: https://arxiv.org/abs/1511.06348 874KB. 2016

[DLH+16] J. Dai, Y. Li, K. He, J. Sun, R-FCN: Object Detection via Region-based Fully Convolutional Networks, URL: https://arxiv.org/abs/1605.06409 8,6MB 2016

[EGW+10] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes (VOC) Challenge, URL: http://homepages.inf.ed.ac.uk/ckiw/postscript/ijcv_voc09.pdf 7,7 MB, 2010

[HZR+15] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, URL: https://arxiv.org/abs/1512.03385 800KB, 2015

[KSH12] A. Krizhevsky, I. Sutskever, G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks URL https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf 1,3MB, 2012

[LBH15] Y. LeCun , Y. Bengio , G. Hinton, Deep learning URL: https://www.nature.com/articles/nature14539 1,98MB 2015

[Per13] A. K. Perzanowski, Tattoos & IP Norms, URL: https://scholarlycommons.law.case.edu/cgi/viewcontent.cgi?article=1046&context=faculty_publications 1,68MB 2013

[SFM+14] E. Simo-Serra, S. Fidler, F. Moreno-Noguer, R. Urtasun, A High Performance CRF Model for Clothes Parsing, URL: https://www.cs.toronto.edu/~urtasun/publications/simo_et_al_accv14.pdf 8,00MB 2014

[ZCD+17] H. Zhang, M. Cisse, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond Empirical Risk Minimization, URL: https://arxiv.org/pdf/1710.09412.pdf 2017

[ZHZ+19] Z. Zhang, T. He, H. Zhang, Z. Zhang, J. Xie, M. Li, Bag of Freebies for Training Object Detection Neural Networks, URL: https://arxiv.org/pdf/1902.04103v1.pdf?fbclid=IwAR1S3lBt_U9BQbzm5uVLlIX15aF1D8nyWDbp_FRk2gPFNHlEdfzFFiulsB4, 2019