

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ BAKALAURO STUDIJŲ PROGRAMA

Neuroninių tinklų naudojimas plaučių ligų analizei

Usage Of Neural Networks In Analysis Of Lung Diseases

Bakalauro baigiamasis darbas

Atliko:	Vidmantas Bakštys	(parašas)
Darbo vadovas:	asist. dr. Tomas Plankis	(parašas)
Darbo recenzentas:	prof. dr. Olga Kurasova	(parašas)

Vilnius – 2020

Santrauka

Darbe nagrinėjamos skirtingos konvoliucinių neuroninių tinklų architektūros, galinčios spręsti plaučių auglių klasifikavimo uždavinį. Taip pat tikrinama šių skirtingų architektūrų neuroninių tinklų bendro naudojimo galimybė. Teorinėje dalyje yra apžvelgiami 3 moksliniuose straipsniuose pasiūlyti konvoliuciniai tinklai - modifikuotas VGG16-CNN tinklas [GZT⁺], Mask-R-CNN tinklas [ZCZ⁺19] bei MGI-CNN tinklas [KYC⁺19].

Bendro skirtingų tinklų naudojimo drauge galimybė egzistuoja tik tada, jei neuroniniai tinklai turi panašią struktūrą, t.y. gali priimti tokią pačią duomenų įvestį ir pateikti klasifikavimo rezultatus tokiu formatu, kuris galėtų būti naudojamas drauge su kitų neuroninių tinklų rezultatais. Remiantis teorinėje dalyje apžvelgtų neuroninių tinklų modeliais ir idėjomis, eksperimentinėje darbo dalyje sukurti 3 skirtingų architektūrų tinklų modeliai, esantys pajėgūs spręsti klasifikavimo uždutį drauge, remdamiesi daugumos ir svoriniu konsensusu.

Darbe lyginamos šių tinklų veikimo ypatybės ir eksperimentiniai rezultatai, taip pat tikrinama prielaida, ar konsensusu besiremiantis nuotraukų klasifikavimas duoda geresnius rezultatus nei pavienių neuroninių tinklų rezultatai.

Raktiniai žodžiai: konvoliuciniai neuroniniai tinklai, neuroninių tinklų konsensusas, VGG16-CNN, Mask-R-CNN, MGI-CNN

Summary

This work analyses different architectural styles of convolutional neural networks that can be used to tackle the problem of lung nodule classification. The possibility of using these networks of different architectures together is also being examined. The theoretical part reviews 3 different networks that were proposed in scientific articles - modified VGG16-CNN network [GZT⁺], Mask-R-CNN network [ZCZ⁺19] and MGI-CNN network [KYC⁺19].

The possibility of using different network architectures together is only possible when the networks have a similar structure - they need to be able to take the same input and produce classification output results in such format that could be used together with outputs of other neural networks. Based on the neural network models and ideas reviewed in the theoretical part, 3 experimental network models of different architectures were developed in the experimental part of the work, which are able to solve the classification task together based on majority and weighted consensus.

This work compares the performance characteristics and experimental results of these networks, as well as examines the assumption whether consensus-based classification of images gives better results than the classification results of individual neural networks.

Keywords: convolutional neural networks, neural network consensus, VGG16-CNN, Mask-R-CNN, MGI-CNN

TURINYS

IVADAS	4
1. DIRBTINIAI NEURONINIAI TINKLAI	6
1.1. Dirbtinio neuroninio tinklo sluoksniai	6
1.2. Dirbtinio neuroninio tinklo mokymas	6
1.3. Neuroninio tinklo aktyvacijos funkcijos	7
1.4. Konvoliuciniai neuroniniai tinklai	7
1.5. Dirbtinių neuroninių tinklų rezultatų įvertinimo charakteristikos	8
2. MOKSLINIUOSE STRAIPSNIUOSE PASIŪLYTŲ TINKLŲ ANALIZĖ	10
2.1. VGG16-CNN tipo konvoliuciniai neuroniniai tinklai	10
2.1.1. Išplėsta konvoliucija	10
2.1.2. Hyperstulpelis	11
2.2. MGI-CNN tipo konvoliuciniai neuroniniai tinklai	11
2.3. MASK-R-CNN tipo konvoliuciniai neuroniniai tinklai	12
3. KONVOLIUCINIŲ NEURONINIŲ TINKLŲ PALYGINIMO EKSPERIMENTAS	14
3.1. Modifikuoti aprašytų konvoliucinių tinklų modeliai	14
3.1.1. „Paprastas“ konvoliucinis neuroninis tinklas	14
3.1.2. Modifikuotas VGG16 konvoliucinis tinklas su svoriniais balsavimais	15
3.1.3. MGI-CNN tinklas	16
3.2. LUNA16 plaučių nuotraukų duomenų bazė	16
3.3. Mašininio mokymosi karkasas „PyTorch“	17
4. KONVOLIUCINIŲ NEURONINIŲ TINKLŲ KONSENSUSAS - JUNG TINIO KLASIFIKA- VIMO METODAI	18
4.1. Daugumos balsavimas	18
4.2. Balsavimas su svoriais	18
4.3. Konsensuso implementavimas praktinėje darbo dalyje	19
5. EKSPERIMENTINIO TYRIMO BANDYMAI IR SUKURTŲ TINKLŲ REZULTATAI	20
5.1. „Paprasto“ konvoliucinio neuroninio tinklo rezultatai	20
5.2. MGI-CNN tipo konvoliucinio tinklo rezultatai	20
5.3. VGG16-CNN tipo konvoliucinio tinklo rezultatai	21
5.4. Konsensuso rezultatai	21
5.4.1. Daugumos balsavimas	21
5.4.2. Balsavimas su svoriais	21
6. SUKURTŲ TINKLŲ PALYGINIMAS IR ĮVERTINIMAS	23
DARBO REZULTATAI	24
IŠVADOS	25
LITERATŪRA	26
SAVOKŲ APIBRĖŽIMAI	27
PRIEDAI	27
1 priedas. Praktinės darbo dalies naudojimo instrukcijos	28

Įvadas

Plaučių vėžys yra viena iš dažniausiai sutinkamų vėžio formų - tai yra dažniausia vėžinių susirgimų rūšis tarp vyrų ir trečia pagal dažnumą tarp moterų[19]. Dideli gydytojų radiologų darbo krūviai apsunkina ankstyvąją ligos diagnozę, todėl yra būtina ieškoti kitų būdų plaučių augliams aptikti. Vienas iš moderniausių ir patikimiausių būdų ligai diagnozuoti - skaitmeninių plaučių nuotraukų analizavimas ir galimų auglių klasifikavimas dirbtiniais neuroniniais tinklais.

Vis didėjantis skaitmeninių medicininių duomenų kiekis verčia ieškoti naujų būdų jų apdorojimui, todėl šiai problemai spręsti vis dažniau pasitelkiami mašininio mokymosi metodai. Mokslinėje literatūroje yra pateikiama įvairių stilių konvoliucinių neuroninių tinklų architektūrų, tačiau ne visi jie yra dažnai sutinkami praktikoje. Teoriškai pasiūlyti tinklai dažnai remiasi įdomiomis idėjomis, tačiau šių idėjų efektyvumas lyginant su dažniau sutinkamomis tinklų architektūromis dažnai lieka neaiškiu. Naujų idėjų išbandymas eksperimentiškai ir jų įvertinimas yra viena iš šio darbo užduočių.

Darbo tikslas - apžvelgti moksliniuose straipsniuose [GZT⁺][ZCZ⁺19][KYC⁺19] publikuotas skirtingas konvoliucinių neuroninių tinklų architektūras, pabandyti patikrinti jų veikimo principus ir idėjas eksperimentiškai bei palyginti jų veikimo rezultatus.

Taip pat siekiama ištirti skirtingų tipų neuroninių tinklų bendro veikimo kartu galimybę.

Darbai pasiruošti buvo atliktas kursinis darbas, kurio metu buvo apžvelgta 10 skirtingų mokslinių straipsnių, kuriuose pristatomi skirtingi plaučių ligų diagnozavimo būdai. Iš šių straipsnių buvo pasirinkti 3 straipsniai, kuriuose aprašyti neuroninius tinklus ar jų idėjas ir stengiasi įgyvendinti šio darbo praktinėje dalyje. Ne visi tinklai yra tinkamai suformuluoti darbo užduočiai - plaučių auglių atpažinimo, t.y. skaitmeninio ligos diagnozavimo, užduočiai spręsti. VGG16 tipo konvoliucinį tinklą aprašantis straipsnis remiasi hyperstulpelio metodu, kuris yra naudojamas plaučių kontūrai nustatyti. Tačiau hyperstulpelio idėją šio darbo autorius bando pritaikyti ir nuotraukų klasifikavimo užduočiai.

Mask-R-CNN tipo konvoliucinis tinklas remiasi gydytojų profesionalų anotuotomis nuotraukomis, kuriose yra pažymėtas plaučių kontūras. Tokio tipo nuotraukos nėra prieinamos viešai, todėl tinklas nėra tinkamas darbo užduočiai. Tačiau Mask-R-CNN tinklas remiasi konsensuso principu, kurį galima pritaikyti ir kitokio tipo tinklų rezultatams sujungti. Šio tinklo konsensuso idėją darbo autorius naudoja kartu su savo įgyvendintais konvoliuciniais neuroniniais tinklais.

Eksperimentinė darbo dalis įgyvendinta Python programavimo kalba (versija 3.8), naudojant jos distribuciją Anaconda (versija 2020.2) bei mašininio mokymosi karkasą PyTorch (versija 1.50). Tinklų įvesties duomenys yra dvimatės kompiuterinės tomografijos plaučių nuotraukos (65x65 pikselių dydžio), gautos programiškai apdorojus LUNA16 kompiuterinės tomografijos plaučių nuotraukų duomenų bazės duomenis.

Darbe nėra siekiama rasti maksimalaus tinklų tikslumo - kadangi tinklai yra skirtingų architektūrų, jų palyginimas yra sudėtingas, t.y. sunku įvertinti ar jų sudėtingumas (gilumas, ištrainavimo lygis ir pan.) yra analogiški. Todėl parinkti tokie tinklų parametrai, kurie leistų tinklui pasiekti pozityvų klasifikavimo rezultatą, bet labiausiai siekta išlaikyti kertines tinklų architektūrų idėjas, t.y. stengiamasi "nepaslėpti" ar paversti nereikšmingomis tų jų detalių, kurios ir daro juos

unikaliais, vardan geresnio klasifikavimo rezultato.

Praktinėje darbo dalyje laikomasi principo, kad visi tinklai turi būti treniruojami ir testuojami su ta pačia įvesties duomenų imtimi. Tokiu būdu tinklus yra lengviau palyginti, taip pat tai leidžia juos kartu naudoti analizuojant tą pačią duomenų įvestį. Norint sužinoti daugiau apie praktinės darbo dalies veikimą ir jos paleidimo instrukcijas, žr. priedus (nr. 1)

Šiame darbe iš pradžių pristatoma teorinės žinios apie neuroninius tinklus, vėliau pristatomos mokslinėje literatūroje aprašytos unikalios konvoliucinių neuroninių tinklų architektūros, o galiausiai pristatomi autoriaus sukurti neuroniniai tinklai, besiremiantys mokslinėje literatūroje aprašytų tinklų architektūra. Darbo dėstymą užbaigia konsensuso mechanizmų aptarimas ir palyginimas.

Rezultatų skyriuje palyginami visų trijų praktiškai įgyvendintų tinklų veikimo rezultatai, taip pat pateikiami skirtingų tinklų konsensuso rezultatai. Rezultatai yra įvertinami, nurodant galimas jų priežastis.

Išvados autorius įvertina tiek savo, tiek ir moksliniuose straipsniuose pasiūlytų tinklų architektūras, nurodo jų stipriąsias ir silpnąsias vietas.

1. Dirbtiniai neuroniniai tinklai

Dirbtinis neuroninis tinklas yra skaičiavimo sistema, kuri yra įkvėpta biologinės neuronų sistemos. Žmogaus smegenis sudaro vidutiniškai 86 milijardai neuronų, iš kurių kiekvienas turi maždaug po 1000 jungčių su kitais neuronais.

Dirbtiniai neuroniniai tinklai yra tinkami duomenų analizavimui ir jų klasifikavimui tada, kai tradiciniai statistiniai metodai nėra pajėgūs to padaryti. Mokslininkai yra sukūrę įvairių tipų neuroninių tinklų architektūrų, kurios yra gerai tinkamos kažkokioms konkrečioms užduotims spręsti, pvz. saviorganizuojantys žemėlapiai - duomenų klasterizavimui, rekurentiniai tinklai yra geri įvesties duomenų srauto realiu laiku analizavimui ir sekančios duomenų vertės prognozavimui ir t.t.

Vienas iš paprasčiausių neuroninių tinklų - daugiasluoksnis perceptronas - remiasi tiesioginiu įvesties duomenų skleidimu pirmyn per neuronų sluoksnius ir klaidos skleidimu atgal, tokiu būdu atnaujinant (mokant) tinklo svorius. Tokio tipo neuroniniai tinklai yra tinkami nuotraukų analizei, tačiau yra nepatogūs dėl savo didelio svorių skaičiaus.

Praktika rodo, kad nuotraukų analizavimui labai gerai tinka konvoliuciniai neuroniniai tinklai.

1.1. Dirbtinio neuroninio tinklo sluoksniai

Dirbtinį neuroninį tinklą sudaro sluoksnių rinkinys. Savo ruožtu sluoksnius sudaro neuronų grupės, kurios yra sujungtos su prieš juos ir po jų esančių sluoksnių neuronais. Kiekviena iš šių jungčių turi savo svorį, kurio reikšmė yra sudauginama su įeinančia reikšme persiunčiant informaciją sekančiam sluoksniui.

Pradinis sluoksnis visados yra įeities sluoksnis. Jo dydį nusako nagrinėjamų įvesties objektų dydis. Pavyzdžiui, jei įvesties objektas yra nuotrauka, tai pradinis neuroninio tinklo sluoksnis turės tiek neuronų, kiek nuotraukoje yra pikselių.

Paskutinio neuroninio tinklo sluoksnio dydis priklauso nuo klasifikuojamų objektų kiekio. Paskutinio sluoksnio reikšmės, vykstant duomenų perdavimui per neuroninį tinklą, galiausiai virsta klasifikuojamų objektų tikimybėmis (pritaikius aktyvacijos funkciją).

Kiekvienas sluoksnis taip pat turi slenkstį (angl. bias). Slenkstis yra neuroninio tinklo sluoksnio dalis, tačiau nėra sujungtas su prieš jį buvusiais sluoksniais, bet yra sujungtas su po jo esančio neuroninio tinklo sluoksnio neuronais. Slenksčio dydis dažniausiai būna fiksuotas.

1.2. Dirbtinio neuroninio tinklo mokymas

Tipiniu atveju, dirbtinis neuroninis tinklas veikia priimdamas informaciją pradiname savo sluoksnyje ir persiųsdamas ją tolimesnio sluoksnio neuronams, sudauginęs gautą informaciją su kiekvienos konkrečios jungties svoriu ir pritaikęs aktyvacijos funkciją. Šis procesas vyksta tol, kol nepasiekiamas išorinis sluoksnis.

Išoriniame sluoksnyje, pritaikius aktyvacijos funkciją, gaunamas klasifikavimo rezultatas. Jei gautasis rezultatas nesutampa su norimu, tai reiškia, kad neuroninio tinklo svoriai nėra geri - juos būtina pataisyti. Klaida iš išorinio sluoksnio yra paskleidžiama atgal (angl. backpropagation), pakoreguojant visus neuroninio tinklo svorius.

Svorių koregavimo stiprumas priklauso nuo to, koks yra nustatytas mokymosi greičio parametras (angl. learning rate) ir koks buvo klaidos dydis. Jei suklysta smarkiai, svoriai bus keičiami smarkiau, nei kai suklysta nedaug (tai priklauso nuo gradiento reikšmės dydžio).

1.3. Neuroninio tinklo aktyvacijos funkcijos

Kiekvieno neurono reikšmė, suskaičiavus įeinančių reikšmių ir atitinkamų jungčių svorių sandaugų sumą, turi būti persiunčiama toliau. Tačiau prieš ją persiunčiant ši reikšmė yra aktyvuojama - t.y. jai yra pritaikoma aktyvavimo funkcija.

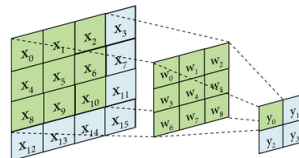
Aktyvavimo funkcijų naudojimas leidžia reikšmėms išlikti nedidelėmis, t.y. jei neuroninis tinklas linkęs įgauti dideles reikšmes, šios reikšmės yra įvedamos į mažesnę intervalą.

Dažniausiai vartojamos aktyvacijos funkcijos:

- Tiesinė - 1, jei $a \geq 0$, ir 0, jei $a < 0$
- Sigmoidinė - suveda reikšmes į intervalą (0;1)
- Softmax - taip pat suveda reikšmes į intervalą (0;1)

1.4. Konvoliuciniai neuroniniai tinklai

Konvoliucinių tinklų skiriamasis bruožas - papildomas tinklo elementas - konvoliucijos sluoksnis. Konvoliucija yra įgyvendinama konvoliucinių filtrų pagalba, transformuojant įvesties sluoksnio duomenis į požymių žemėlapius (žr. 1 paveikslą).



1 pav. Konvoliucija. Per 4x4 dydžio požymių žemėlapi slenkamas 3x3 dydžio filtras, taip sudarant naują, mažesnę požymių žemėlapi. [MDB17]

Konvoliucinis filtras - dažniausiai mažų matmenų duomenų tenzorius, kurio elementai turi savo konkrečią vietą šiame tenzoriuje (jei filtras dvimatis - tuomet jo duomenys sudaro matricą), t.y. tenzorius elementų išdėstymo tvarka yra svarbi.

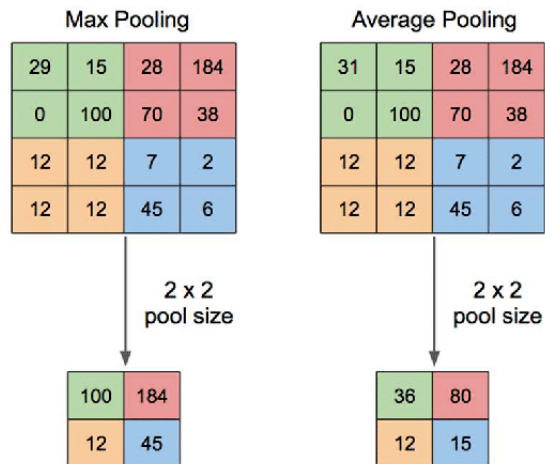
Šis vietos svarbos tenzoriuje principas galioja ir nuotraukoms - svarbūs ne tik patys pikseliai ir jų spalva, tačiau ir jų vieta nuotraukoje, t.y. jų išsidėstymas ir sudaro nuotraukoje matomą paveikslą.

Konvoliucija, kaip tinklo elementas, transformuoja įvesties duomenis į naują požymių žemėlapi (dažniausiai mažesnio dydžio), tačiau šis požymių žemėlapis tiesiogiai priklauso nuo prieš tai buvusio sluoksnio pikselių pozicijų, t.y. naujasis požymių žemėlapis išlaiko didžiąją dalį informacijos apie prieš tai buvusio sluoksnio pikselių išsidėstymą.

Ši savybė ir yra svarbiausia, kodėl konvoliuciniai tinklai yra tokie patogūs nuotraukų skaitmeninei analizei - nepaisant sumažinto įvesties tenzorius dydžio po konvoliucijos, didžioji dalis

informacijos apie prieš tai buvusį sluoksnį išlieka neprarasta. Tai leidžia sumažinti įvesties duomenų kiekį, kurį reikia vėliau analizuoti pilnai sujungtu duomenų sluoksniu (perceptronu), norint gauti galutinį klasifikavimo rezultatą.

Kitas būdas sumažinti įvesties duomenų kiekį - sutraukimo sluoksnio naudojimas (angl. pooling - žr. 2 paveikslą). Dažniausiai naudojami sutraukimo būdai - „max pooling“ ir „average pooling“.



2 pav. Sutraukimas. Iš 2x2 dydžio duomenų matricos sudaroma naujo pikselio reikšmė. [YIS19]

Tipiniu atveju, treniravimo pradžioje konvoliuciniai filtrai yra inicializuojami atsitiktinėmis reikšmėmis, t.y. reikšmės gali būti ir neigiamos. Tokiu atveju ir naujų požymių žemėlapių elementų reikšmės gali tapti neigiamomis, kas nėra paranku juos analizuojant toliau. Kad tokių reikšmių nebūtų, naudojamas papildomas - ReLU - sluoksnis.

Taip pat požymių žemėlapių reikšmės gali įgauti pakankamai didelius skirtumus - tai nėra gerai, tinklas dirba geriau, jei duomenys labai smarkiai nesiskiria vieni nuo kitų. Duomenų išlyginimui dažnai naudojamas normalizacijos sluoksnis (angl. batch normalization), kuris padeda spręsti „sprogstančių gradientų“ problemą [DDQ⁺17].

1.5. Dirbtinių neuroninių tinklų rezultatų įvertinimo charakteristikos

Neuroninių tinklų veikimo rezultatai gali būti įvairūs, t.y. tam tikros tinklų architektūros turi skirtingas tendencijas geriau ar blogiau klasifikuoti kažkokius duomenis. Tai taip pat priklauso ir nuo pradinių duomenų bei pasirinktų tinklo hyperparametrų.

Skirtingo tipo klaida klasifikuojant duomenis gali turėti skirtingas pasekmes. Plaučių auglių analizavimo užduoties atveju, neatpažinti esančio auglio yra žymiai blogiau, nei suklysti ir klasifikuoti nuotrauką kaip teigiamą, nors auglio ir nėra. Binarinio klasifikavimo atveju (t.y. kai klasifikuojama tik į dvi klases - yra arba nėra), klasifikavimo rezultato ir tikrosios diagnozės santykis gali būti išreikštas matrica (žr. 1 lentelę).

		gauta klasė	
		C_1	C_2
tikroji klasė	C_1	tikrai teigiamas (TT)	klaidingai teigiamas (KT)
	C_2	klaidingai neigiamas (KN)	tikrai neigiamas (TN)

1 lentelė. Klasifikavimo matrica [Kur19]

Klasifikavimo matricos sąvokos:

- Tikrai teigiamas - klasifikavimo rezultatas yra teigiamas ir jis yra teisingas
- Tikrai neigiamas - klasifikavimo rezultatas yra neigiamas ir jis yra teisingas
- Klaidingai teigiamas - klasifikavimo rezultatas yra teigiamas ir jis yra neteisingas
- Klaidingai neigiamas - klasifikavimo rezultatas yra neigiamas ir jis yra neteisingas

Klaidingai neigiami rezultatai plaučių auglių analizavimo atveju dažniausiai pasitaiko daug rečiau, nei klaidingai teigiami. Iš šių rezultatų sudaromi kiti neuroninių tinklų rezultatus apibūdinantis rodikliai:

$$\text{jautrumas} = \frac{\text{TT skaičius}}{\text{TT skaičius} + \text{KN skaičius}},$$

$$\text{specifiškumas} = \frac{\text{TN skaičius}}{\text{TN skaičius} + \text{KT skaičius}},$$

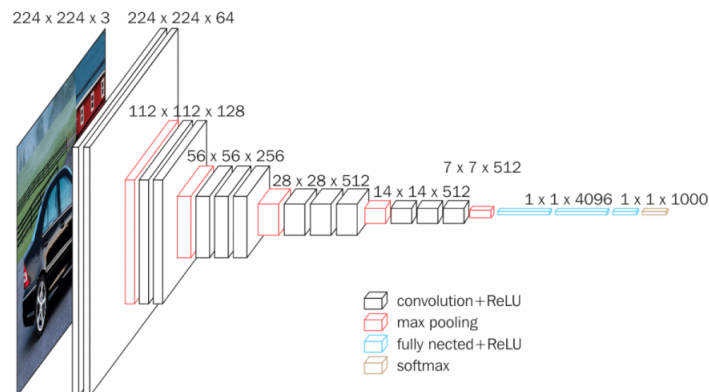
$$\text{bendras klasifikavimo tikslumas} = \frac{\text{TT skaičius} + \text{TN skaičius}}{\text{visų objektų skaičius}}$$

2. Moksliniuose straipsniuose pasiūlytų tinklų analizė

Šiame skyriuje aprašomi trijų neuroninių tinklų architektūros, aprašytos moksliniuose straipsniuose. Visi 3 tinklai remiasi originaliais principais, todėl yra verti atskiro aptarimo.

2.1. VGG16-CNN tipo konvoliuciniai neuroniniai tinklai

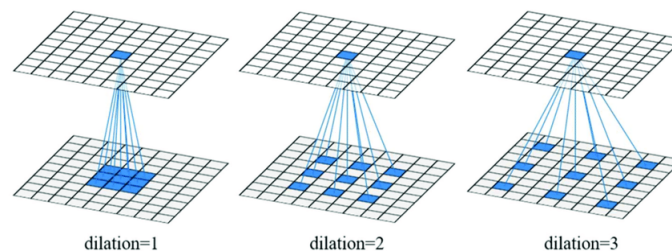
VGG16 tipo konvoliucinis tinklas pirmą kartą buvo pasiūlytas 2014 m. Oksfordo universiteto mokslininkų [SZ14]. Originalus VGG16 tinklas turi 13 konvoliucijos sluoksnių ir 3 pilnai sujungtus sluoksnius. Įvesties dydis yra 224×224 pikselių dydžio. Daug konvoliucijos sluoksnių leidžia sutraukti požymių žemėlapių dydį prieš analizuojant duomenis pilnai sujungtais sluoksniais, tačiau VGG16 tinklas dėl savo didelio sluoksnių skaičiaus yra žinomas kaip lėtai treniruojamas tinklas (žr. 3 paveikslą).



3 pav. Originalaus VGG16 tinklo architektūra [Hus]

2.1.1. Išplėsta konvoliucija

Nagrinėjamo mokslinio straipsnio [GZT⁺] autoriai siūlo pakeisti paskutinius konvoliucinius sluoksnius kitokio tipo konvoliucijomis - išplėstomis konvoliucijomis (žr. 4 paveikslą).



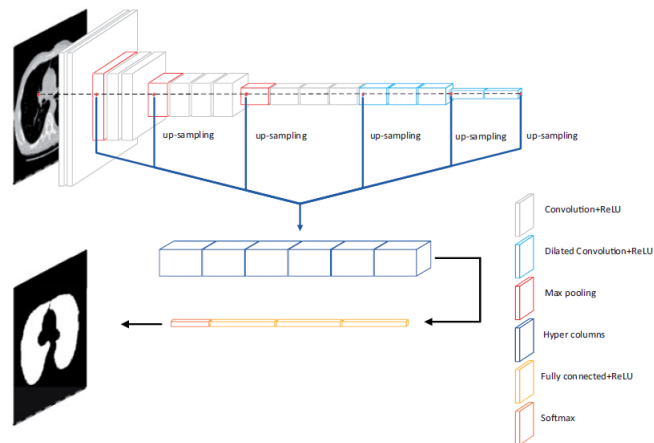
4 pav. Išplėsta konvoliucija [CZG⁺19]

Išplėstos konvoliucijos leidžia aprėpti platesnį požymių žemėlapių diapazoną, tačiau filtro dydis dėl to netampa dideliu, t.y. keičiant išplėtimo lygį galima didinti aprėpiamą diapazoną nekeičiant filtro dydžio.

2.1.2. Hyperstulpelis

Straipsnyje nagrinėjama plaučių kontūro nustatymo problema. Sprendžiant tokio tipo užduotį yra vykdomas kiekvieno nuotraukos pikselio klasifikavimas, t.y. ar jis yra plaučių pikselis, ar fonas.

Hyperstulpelis - pikselio reprezentacijų masyvas skirtinguose tinklo sluoksnių gyliuose. Vykdamant konvoliucijas, požymių žemėlapių dydis mažėja, tačiau naudojant interpoliaciją galimas požymių žemėlapių grąžinimas į pradinį dydį (angl. upsampling). Tokiu atveju, kiekviename tinklo sluoksnyje, naudojant nuotraukos dydžio grąžinimą į pradinį dydį, kiekvienas pikselis turi savo atitikmenį atitinkamame požymių žemėlapyje (žr. 5 paveikslą).



5 pav. Siūloma VGG16 tinklo modifikacija - hyperstulpelis analizuojamas perceptrono. [GZT⁺]

Hyperstulpelio idėja leidžia pasiekti ankstesnių tinklo sluoksnių duomenis, kurie dar nėra taip smarkiai „perdirbti“ tinklo. Tai praturtina modelį platesne informacija.

Ši idėja pirmą kartą buvo pristatyta 2019 m., tiriant neuroninių tinklų naudojimą smegenų magnetinio rezonanso nuotraukų analizėje [TCE20]

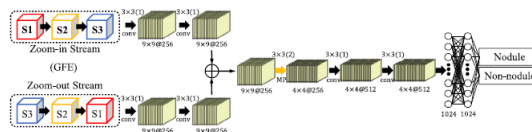
2.2. MGI-CNN tipo konvoliuciniai neuroniniai tinklai

MGI-CNN tipo konvoliucinis tinklas remiasi dviejų nuotraukų analizavimo srautų - pritraukiant nuotrauką ir atitraukiant nuotrauką - analize. Abu srautai analizuojami lygiagrečiai, o vėliau sujungiami pilnai sujungtu sluoksniu.

Iš pradžių iš to pačio nuotraukų rinkinio yra sudaromi 3 nuotraukų rinkiniai - 40x40 pikselių, 30x30 pikselių ir 20x20 pikselių dydžio. Visi šie rinkiniai turi tą patį centrą, tačiau mažesnės nuotraukos yra „apkirptos“. Tada visos nuotraukos yra sumažinamos taip, kad būtų 20x20 pikselių dydžio (interpoliacijos pagalba). Tokiu atveju gaunami vienodų dydžių nuotraukų rinkiniai, bet su skirtingu vaizdo pritraukimu.

Nuotraukų analizavimo pritraukiant (angl. zoom in) srautas iš pradžių taiko konvoliuciją 40x40 (sumažintai interpoliacijos pagalba iki 20x20) pikselių dydžio nuotraukai. Gauti požymių žemėlapiai yra konkatenuojami kartu su 30x30 (o iš tiesų - 20x20 pikselių) nuotraukų įvestimi. Tada šiai įvesčiai yra vėl taikoma konvoliucija ir gauti požymių žemėlapiai konkatenuojami su 20x20 pikselių dydžio nuotraukos įvestimi. Galiausiai visiems požymių žemėlapiams yra bendrai taikoma

konvoliucija, o tada abu srautai (pritraukimo ir atitraukimo) sujungiami pilnai sujungtu sluoksniu (žr. 6 paveikslą).



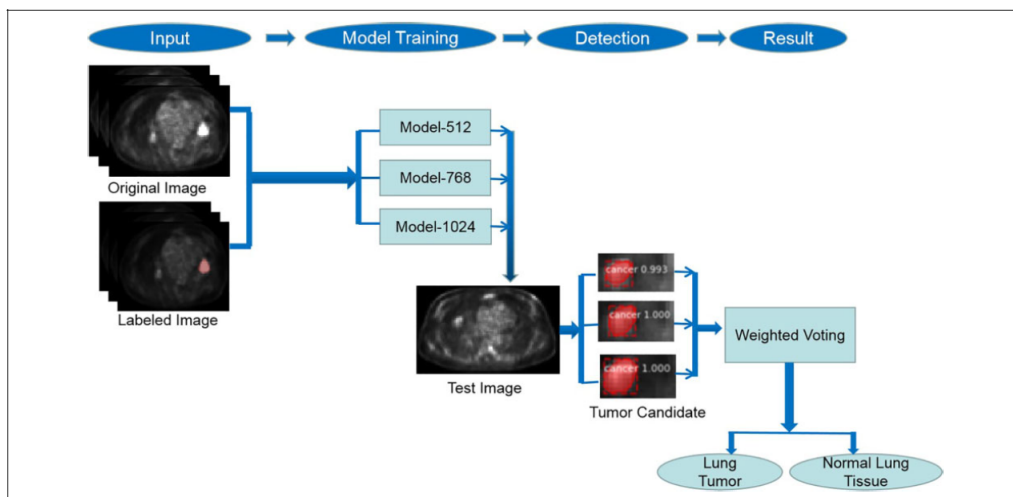
6 pav. MGI-CNN konvoliucinio neuroninio tinklo modelis [KYC⁺ 19]

Atitraukimo srautas (angl. zoom out) veikia analogiškai, tik pradeda konvoliucijas taikyti 20x20 pikselių dydžio nuotraukai, o užbaigia konvoliucijomis su 40x40 pikselių nuotraukomis.

Kaip buvo minėta, visų nuotraukų dydžiai buvo suvienodinti ruošiant duomenų rinkinius. Tačiau vykdant konvoliucijas, nuotraukų dydis sumažėja, t.y srauto pradžioje analizuojamos nuotraukos dydis tampa mažesniu. Tačiau jis vėl atstatomas interpoliacijos metodu, kad nuotraukų dydžiai išliktų 20x20 pikselių dydžio visą tinklo treniravimo laiką.

2.3. MASK-R-CNN tipo konvoliuciniai neuroniniai tinklai

Mask-R-CNN tipo tinklas yra naudojamas plaučių auglių aptikimui. Tinklas naudoja R-CNN konvoliucinį tinklą, kaip savo pagrindą, tačiau prideda jam papildomo funkcionalumo. Tinklas analizuoja nuotrauką, lygindamas jos atskirus regionus (angl. ROI - region of interest) su iš anksto anotuotomis nuotraukomis, kuriose yra pažymėtas plaučių auglio kontūras. „Uždengti“ anotuoto regiono analizuojamos nuotraukos pikseliai toliau yra analizuojami R-CNN tinklo, tokiu būdu klasifikuojant auglio buvimą ar nebuvimą būtent tame regione (žr. 7).



7 pav. Mask-R-CNN konvoliucinio neuroninio tinklo modelis [ZCZ⁺ 19]

Tinklas taip pat taiko konsensuso mechanizmą auglių atpažinimui. Ta pati „kaukė“ tinklo treniravimo metu yra išgaunama iš tos pačios nuotraukos, bet pateiktos trimis skirtingomis rezoliucijomis. Klasifikuojant taikomų kaukių atitikimą uždengtam plotui, visos 3 kaukės vykdo svorinį balsavimą. Balsavimo rezultatas laikomas teigiamu, jei šių kaukių ir jų svorių sandaugų suma yra didesnė už nustatytą apatinę ribą.

Straipsnio autoriai teigia, kad balsavimo metodo naudojimas padeda sumažinti klaidingai teigiamų klasifikavimo rezultatų skaičių.

3. Konvoliucinių neuroninių tinklų palyginimo eksperimentas

Eksperimento tikslas - sukurti 3 neuroninius tinklus ir patikrinti nagrinėtuose straipsniuose aprašytų konvoliucinių neuroninių tinklų architektūras - arba juose taikomas idėjas.

Ne visi tinklai pilnai tinka nagrinėjamai užduočiai - pvz. Mask-R-CNN konvoliucinis neuroninis tinklas reikalauja kitokio tipo - su pažymėtu kontūru - plaučių auglių nuotraukų. Praktiniame darbe nagrinėjama duomenų bazė tokių duomenų nepateikia, ji tik nurodo nagrinėjamos nuotraukos klasę.

Mask-R-CNN tipo tinklas yra pakeistas „paprastu“ konvoliuciniu neuroniniu tinklu, turinčiu labiau tradicinę struktūrą, tačiau Mask-R-CNN tinklo idėja naudoti balsavimą priimant galutinį klasifikavimo sprendimą yra naudojama vykdant skirtingų konvoliucinių neuroninių tinklų konsensą, taip pat yra naudojama modifikuotoje VGG16 tinklo versijoje.

Tinklai paruošti taip, kad būtų aiškos struktūros ir jų treniravimas neužtruktų pernelyg ilgai, nors praktikoje jis vis tiek yra gana ilgas. Jei tokio tipo tinklai būtų naudojami produkcinėje aplinkoje, tikėtina, kad konvoliucinių filtrų ar net konvoliucijos sluoksnių didesnis skaičius galėtų duoti geresnius klasifikavimo rezultatus, tačiau tai nėra šio darbo tikslas.

Darbe apsiribojama pakankamai nedideliais modeliais, kurie aiškiai parodo esminius architektūros veikimo principus ir efektyvumą.

3.1. Modifikuoti aprašytų konvoliucinių tinklų modeliai.

Tolesniuose poskyriuose aprašomos visų trijų eksperimentui paruoštų konvoliucinių neuroninių tinklų architektūros.

3.1.1. „Paprastas“ konvoliucinis neuroninis tinklas

Tinklas susideda iš 6 konvoliucinių sluoksnių, kuriuose yra po 32 konvoliucinius filtrus. Po kiekvienos konvoliucijos požymių žemėlapiams yra taikoma duomenų normalizacija (angl. batch normalization). Po 3 ir 6 konvoliucijų yra vykdomas požymių žemėlapių sutraukimas (angl. max pooling). Galiausiai požymių žemėlapiai analizuojami pilnai sujungto sluoksnio.

Tinklas naudoja Adam optimizatorių, o praradimui apskaičiuoti naudojamas kryžminės entropijos praradimų skaičiavimo algoritmas. Klasių klasifikavimo įverčiams gauti naudojama Soft-max funkcija.

Tinklo naudojamoms konvoliucijoms leidžiama mažinti požymių žemėlapius (kadangi tinklo architektūra yra paprasta, nėra pagrindo elgtis kitaip), todėl šis tinklas yra gana greitai treniruojamas.

Šis modelis yra pats paprasčiausias iš visų 3 implementuotų modelių, todėl yra patogus naudoti kitų tinklų palyginimui, kaip etaloninis konvoliucinis tinklas.

3.1.2. Modifikuotas VGG16 konvoliucinis tinklas su svoriniais balsavimais

Ankstesniame skyriuje aprašytas VGG16 tinklas (žr. 2.1 skyrių) yra skirtas plaučių kontūro nustatymui klasifikuojant kiekvieną nuotraukos pikselį, todėl nėra visai tinkamas plaučių auglių klasifikavimui. Nepaisant to, hyperstulpelio panaudojimas yra esminis modelio bruožas, todėl nuspręsta jį išlaikyti.

Tokia hyperstulpelio interpretacija, kokia yra aprašyta straipsnyje, duoda prastus klasifikavimo rezultatus, todėl nuspręsta ją šiek tiek pakeisti. Straipsnyje aprašytame modelyje hyperstulpeliu laikomi pikselio atitikmenys skirtinguose šešiuose tinklo gyliuose. Tada šie 6 duomenys yra paduodami pilnai sujungtam tinklui kaip įvestys ir gaunamas pikselio klasifikavimo rezultatas.

Tačiau eksperimentinės užduoties tikslas yra klasifikuoti ne pikselius, o plaučių auglius. Todėl būtina išlaikyti nuotraukos vientisumą, t.y. jos negalima karpyti po pikselį ir nagrinėti atskirai.

Sugalvotas sprendimo būdas - panaudoti skirtingų gylių konvoliucijos sluoksnių duomenis (kaip ir hyperstulpelyje) analizuojant nuotrauką, tačiau šiuos duomenis analizuoti ne vienu pilnai sujungtu sluoksniu, o naudoti po naują pilnai sujungtą sluoksnį kiekvienam hyperstulpelio sluoksniui atskirai.

Tokiu būdu gaunami „skirtingo gylio“ tinklo rezultatai - pradedant nuo „mažo“ tinklo iki „didelio“ tinklo, t.y. tinklo, turinčio visus sluoksnius.

Tinklo treniravimas susideda iš kelių etapų. Iš pradžių tinklas treniruojamas taip, tarsi jis būtų tiesiog originalaus VGG16 stiliaus konvoliucinis neuroninis tinklas (tačiau turintis išplėstinių konvoliucijų paskutiniuose dviejuose tinklo sluoksniuose, kaip ir pasiūlyta moksliniame straipsnyje), t.y. pilnai išnaudojami visi tinklo sluoksniai, o tinklo parametrų svoriai yra mokomi.

Ištreniravus tinklą, visų konvoliucinių filtrų svoriai yra užrakinami ir toliau lieka nebepakitę.

Tada treniruojamas kiekvieno iš hyperstulpelio sluoksnių pilnai sujungtas sluoksnis (t.y. perceptronas, prijungtas prie kažkurios iš tinklo konvoliucijų). Kiekvieno ištreniruoto hyperstulpelio perceptrono treniravimo tikslumas yra išsaugomas tinkle.

Testuojant tinklą, pirmasis žingsnis, vykdytas treniravime, nebėra vykdomas. Naudojami tik visi 6 hyperstulpelio perceptronai, grąžinantys klasifikavimo rezultatą.

Galutinis klasifikavimo sprendimas yra priimamas vykdant svorinį balsavimą tarp visų šešių hyperstulpelio perceptronų. Svorinis balsavimas vykdomas sudauginant kiekvieno perceptrono rezultatą, pritaikius jam Softmax funkciją, su šio hyperstulpelio perceptrono tikslumo istorijos įverčiu, ir visas šias reikšmes sudedant.

Ši tinklo architektūra su balsavimu leidžia pasiekti panašių rezultatų, kaip ir originalaus VGG16 stiliaus tinklas (toks, koks yra naudojamas treniravimo pirmame etape).

Tinklo treniravimas trunka labai ilgai, nes treniravimas vyksta ne vienu etapu ir vykdoma 7 kartus daugiau treniravimo epochų, nei paprastesnės architektūros tinkluose (nors hyperstulpelio perceptronų treniravimas ir yra greitesnis, nes nebėra keičiami konvoliucinių filtrų svoriai).

Dėl šios priežasties tinklo konvoliucinių filtrų kiekis yra parinktas mažas (po 4 kiekvienam konvoliuciniam sluoksniui). Filtrų dydis 3x3 pikselio.

Po kiekvienos konvoliucijos vykdoma normalizacija (angl. batch normalization), o po antros ir septintos konvoliucijos - sutraukimas (angl. max pooling).

Tinklas naudoja Adam optimizatorių, kryžminės entropijos nuostolio funkciją nuostoliui skaičiuoti bei Softmax funkciją klasifikavimo įverčiams gauti.

3.1.3. MGI-CNN tinklas

Tinklas implementuotas taip, kaip ir aprašyta anksčiau nagrinėtame straipsnyje (žr. 2.2 skyrių), tačiau naudojama mažiau konvoliucinių filtrų konvoliuciniuose sluoksniuose.

Tinklas iš viso turi 14 konvoliucijos sluoksnių, iš kurių 6 yra pritraukimo (angl. zoom in) srauto konvoliuciniai sluoksniai, 6 - atitraukimo (angl. zoom out) srauto konvoliuciniai sluoksniai, o 2 - bendri konvoliuciniai sluoksniai abiem srautams.

Po kiekvienos konvoliucijos požymių žemėlapiai yra normalizuojami (angl. batch normalization).

Tinklas naudoja Adam optimizatorių, kryžminės entropijos nuostolio funkciją nuostoliui skaičiuoti bei Softmax funkciją klasifikavimo įverčiams gauti.

Pradinės įvesties nuotraukų aibės dydis yra iškart apkarpos iki 40 pikselių, nors testinių duomenų aibė ir yra didesnių matmenų (nes nuotraukos vis tiek yra iškerpamos - nėra prasmės laikyti didesnių).

Straipsnyje teigiama, kad tokio tipo tinklas padeda išvengti klaidingai teigiamų klasifikavimo rezultatų, tačiau praktikos rezultatai nerodo tokios tendencijos.

3.2. LUNA16 plaučių nuotraukų duomenų bazė

LUNA16 duomenų bazė buvo sudaryta plaučių auglių skaitmeninio atpažinimo konkursui, kuriame galėjo varžytis įvairios komandos iš viso pasaulio. Ši duomenų bazė yra sudaryta iš kitos duomenų bazės, LIDC-IDRI, duomenų, tačiau pateiktų patogesniu formatu.

Duomenų bazėje duomenys yra pateikti suskaidyti į 10 skirtingų grupių - tokiu būdu leidžiant iškart lengviau atskirti mokymo ir testavimo duomenų aibes. Duomenys yra pateikiami .mhd formato failais, kurie yra surišti su kitais, .raw tipo failais. Savarankiškai skaityti tokio tipo duomenis nėra paprasta, tačiau yra programinių bibliotekų, kurios padeda tai daryti.

Failuose įrašyti duomenys yra trimačiai kompiuterinės tomografijos duomenys. Atskirame faile pateiktos koordinatės, kuriose gali būti auglys (jei nuotraukos klasė yra teigiama). Nurodytos koordinatės nusako patį auglio centrą, todėl šių koordinatinių buvimas leidžia išgauti tokias nuotraukas, kuriose potencialus auglys būtų pačiame nuotraukos centre.

Tai yra svarbu vėliau jas analizuojant, pvz. galima nukirpti nuotraukos kraštus žinant, kad nuotrauka nebus sugadinta.

Pilnai išgaunamų nuotraukų skaičius duomenų bazėje yra virš 700000, o pats duomenų kiekis yra virš 100 GB dydžio. Didžioji dalis nuotraukų yra ne auglio nuotraukos.

Eksperimentui buvo naudota tik nedidelė dalis duomenų. Duomenų aibės sumažinimas buvo būtinas dėl per didelių laiko sąnaudų, taip pat per didelis ne auglių nuotraukų kiekis treniravimo duomenų aibėje neigiamai veikia treniravimo rezultatus.

3.3. Mašininio mokymosi karkasas „PyTorch“

„PyTorch“ yra atviro kodo mašininio mokymosi karkasas Python programavimo kalbai, daugiausiai kuriamas kompanijos „Facebook“ dirbtinio intelekto tyrimų laboratorijos. „PyTorch“ buvo sukurtas kitos bibliotekos - „Torch“ - pagrindu.

Šis karkasas yra puikiai pritaikytas dirbtinių neuroninių tinklų, tame tarpe ir konvoliucinių neuroninių tinklų, kūrimui. Didžioji dalis operacijų, atliekamų su neuroniniais tinklais, yra implementuotos taip gerai, kad paslepia tikrąjį vykdymo eiliškumą.

Tai yra įmanoma dėl to, kad kiekvienas naujas modelis, kuriamas „PyTorch“ pagrindu, yra kuriamas kaip klasė, paveldinti nn.Module klasę. Ši tėvinė klasė turi metodus, kurie leidžia automatiškai sudaryti tinklo struktūros grafą, todėl, pavyzdžiui, klaidos skleidimas atgal vyksta automatiškai - karkasas pats mato, kuriuo eiliškumu vyko duomenų skleidimas pirmyn.

Karkasas pasirinktas eksperimentui dėl savo aiškumo, taip pat dėl galimybės išnaudoti kompiuterio vaizdo plokštę greitesniam tinklų treniravimui.

4. Konvoliucinių neuroninių tinklų konsensusas - jungtinio klasifikavimo metodai

Šiame skyriuje aptariami galimi bendri įėjties duomenų klasifikavimo metodai. Neuroninių tinklų konsensusas buvo nagrinėtas viename iš aptartų straipsnių, tačiau jis buvo naudojamas to paties neuroninio tinklo viduje.

Taip pat konsensusas buvo pritaikytas darbo autoriaus pristatytame VGG16 stiliaus konvoliuciniame neuroniniame tinkle, tačiau jis taip pat buvo taikomas to pačio tinklo viduje, t.y. jis buvo įtrauktas į pačio tinklo architektūrą.

Galimybė priimti sprendimą kartu tarp atskirų tinklų yra akivaizdi - tinklai gali būti jungiami į vieną bendrą tinklą, tačiau svarbu gerai apgalvoti tokio jungimo strategiją. Neuroniniai tinklai privalo būti tarpusavyje suderinami, t.y. gebėti nagrinėti tą pačią įvestį, o klasifikavimo rezultatų sujungimo strategija turi turėti logišką pagrindą.

4.1. Daugumos balsavimas

Pats paprasčiausias tinklų konsensuso metodas - daugumos balsavimas. Tinklų „balsai“ yra vienodos svarbos, todėl nugali tas balsavimo rezultatas, kuris surenka daugiausiai balsų.

Akivaizdu, kad daugumos balsavime gali dalyvauti tik nelyginis neuroninių tinklų skaičius, nes lyginio skaičiaus atveju nebus aišku, koks turėtų būti galutinis klasifikavimo rezultatas.

Tokio balsavimo didžiausias trūkumas yra tai, kad neatsižvelgiama į kiekvieno balsavime dalyvaujančio tinklo stipriąsias puses.

4.2. Balsavimas su svoriais

Skirtingai nei daugumos balsavimas, balsavimas su svoriais neprivalo turėti nelyginio balsų skaičiaus (nebent svoriai yra parinkti neprotingai, o atskirų tinklų klasifikavimo rezultatai yra imami kaip galutiniai, t.y. klasės numerio, duomenys).

Svorių parinkimo strategija gali remtis prielaidomis (vienas tinklas yra geresnis už kitą, todėl jo svoris turėtų būti didesnis) arba empiriniais duomenimis.

Empiriniai duomenys priimant sprendimą, koks svoris turėtų būti priskirtas kuriam tinklui, turi būti arba surinkti iš anksto (treniruojant ar testuojant tinklą), arba nustatomi tada, kai to prireikia.

Svorio nustatymas, kai jo prireikia, nėra geras būdas - neuroninio tinklo efektyvumo įvertinimas galimas tik tada, jei su juo apdorojama didelis kiekis duomenų ir gaunami jo efektyvumo rezultatai. Neturint svorių iš anksto reikėtų, kad tinklas turi atlikti papildomą testavimosi žingsnį kiekvieną kartą vykdydamas klasifikavimo uždavinį.

Neuroniniai tinklai gali turėti skirtingas stipriąsias ir silpnąsias puses - pvz. vienokios architektūros tinklas geriau atpažįsta klaidingai teigiamus rezultatus nei kitokios architektūros tinklas, todėl jo svoris turėtų būti didesnis, jei bendras klasifikavimo rezultatas yra teigiamas, ir reikia patikrinti, ar nebuvo suklysta.

Taip pat galima „pasikonsultavimo“ strategija, pvz. tinklas priėmė sprendimą, kad nagrinėjamos nuotraukos klasė - teigiama, tačiau žinome, kad klaidingai teigiamo klasifikavimo dažnumas yra pakankamai didelis, todėl reikalingas konsensuso balsavimas su kitu tinklu.

4.3. Konsensuso implementavimas praktinėje darbo dalyje

Praktinėje darbo dalyje konsensuso mechanizmas įgyvendintas dviem būdais:

- Modifikuotas VGG16 konvoliucinis tinklas priima sprendimą konsensuso pagrindu tarp skirtinguose tinklo gyliuose ištreniuotų perceptronų (žr. 3.1.2 skyrių)
- Praktinėje dalyje suprogramuotas mechanizmas, leidžiantis naudoti ištreniuotus ir išsaugotus diske visų trijų suprogramuotų stilių konvoliucinius neuroninius tinklus priimant galutinį sprendimą.

Suprogramuotas konsensuso mechanizmas leidžia naudoti tiek daugumos, tiek ir svoriais paremtą strategiją.

Svoriais paremta strategija remiasi ne galutinių klasifikavimo rezultatų (t.y. teigiamo arba neigiamo rezultato), o tinklo Softmax funkcijos (prieš galutinį klasifikavimą) rezultatu su abiejų klasių įverčiais sudauginimu su tinklo patikimumo įverčiu.

Patikimumo įvertis nustatomas dar testuojant tinklą, todėl tik ištestuoti tinklai gali dalyvauti neuroninių tinklų konsensuso balsavime.

Galiausiai visi šie įverčiai sudedami tarpusavyje ir nustatoma galutinė klasė.

5. Eksperimentinio tyrimo bandymai ir sukurtų tinklų rezultatai

Kuriant ir jau sukūrus konvoliucinių neuroninių tinklų modelius, šie modeliai buvo testuojami. Vienareikšmiškai lyginti tinklus yra sudėtinga, kadangi skiriasi jų mokymosi greičiai (pvz. VGG16 tipo tinklui reikia daug epochų, kad jis rodytų gerus rezultatus) ir architektūros, nepaisant to, kad tinklai yra suderinti pagal savo įvestį ir išvestį.

Svarbu paminėti, kad visi sukurti tinklai naudoja duomenų įkrovimą į atmintį su atsitiktiniu duomenų išmaišymu, todėl bandant atkartoti eksperimentą galima tikėtis panašių, bet ne visiškai tokių pat rezultatų.

Eksperimentui naudojama dalis LUNA16 nuotraukų kolekcijos - iš viso 7901 nuotraukos, iš kurių 1556 yra teigiamos, o likusios - neigiamos. Nuotraukos yra išskirstytos į 10 atskirų apylygių grupių, iš kurių 8 grupės yra naudojamos tinklų mokymui, 1 - testavimui, kurio pagalba nustatomi ir išsaugomi tinklų tikslumai, ir likusi 1 - niekaip nedalyvaujanti mokyme. Pastaroji duomenų grupė ir yra naudojama eksperimentiniams tyrimams.

5.1. „Paprasto“ konvoliucinio neuroninio tinklo rezultatai

„Paprastas“ konvoliucinis tinklas, naudojamas kaip etalonas abiejų likusių tinklų palyginimui, rodo pakankamai gerus treniravimo rezultatus. Jis labai retai klysta nustatydamas klaidingai neigiamą klasifikavimo rezultatą, tačiau kartais klasifikuoja nuotraukas klaidingai kaip teigiamas (žr. 2 lentelę).

Treniravimo epochų skaičius	Tikslumas	Klaidingai teigiami	Klaidingai neigiami
100	0,863	0,121	0,016
50	0,861	0,126	0,013
20	0,875	0,104	0,012
19	0,842	0,151	0,007

2 lentelė. „Paprasto“ konvoliucinio neuroninio tinklo rezultatai

5.2. MGI-CNN tipo konvoliucinio tinklo rezultatai

MGI-CNN tipo tinklas rodo teigiamus rezultatus (apytiksliai 70% atpažinimo tikslumą), tačiau atsilieka nuo abiejų likusių tinklų savo tikslumu.

Įdomu pastebėti, kad šis tinklas buvo implementuotas tiksliausiai laikantis nurodymų iš visų moksliniuose straipsniuose nagrinėtų tinklų, tačiau praktiniai rezultatai rodo, kad jo klaidingai teigiamų klasifikavimų skaičius yra pats didžiausias iš visų tinklų.

Ilgesnis tinklo treniravimas neduoda teigiamų rezultatų, t.y. tinklo tikslumas nusistovi maždaug ties 70% riba nepaisant to, kiek epochų jį betreniruotume (žr. 3 lentelę).

Treniravimo epochų skaičius	Tikslumas	Klaidingai teigiami	Klaidingai neigiami
100	0,717	0,280	0,003
70	0,701	0,297	0,003

3 lentelė. MGI-CNN stiliaus konvoliucinio neuroninio tinklo rezultatai

5.3. VGG16-CNN tipo konvoliucinio tinklo rezultatai

VGG16-CNN tipo tinklas rodo teigiamus rezultatus, tačiau itin ilgas jo treniravimo laikas neleidžia jo nuodugniau panagrinėti (žr. 4 lentelę).

Treniruojant jį su mažesne duomenų aibe, nei nurodytoji šiame eksperimente, jis rodė labai panašius rezultatus į „paprasto“ konvoliucinio neuroninio tinklo. Bandymo duomenys su šia duomenų testavimo seka gali būti neiliustratyvūs (žr. 4 lentelę)

Treniravimo epochų skaičius	Tikslumas	Klaidingai teigiami	Klaidingai neigiami
100	100	0	0

4 lentelė. VGG16-CNN stiliaus konvoliucinio neuroninio tinklo rezultatai

5.4. Konsensuso rezultatai

Panagrinėkime abudu konsensuso atvejus - daugumos balsavimą ir balsavimą su svoriais.

5.4.1. Daugumos balsavimas

Daugumos balsavimo eksperimentiniai rezultatai pademonstruoti lentelėje (žr. 5 lentelę).

Tinklas)	Tinklas	Tinklas	Tikslumas	Klaidingai teigiami	Klaidingai neigiami
CNN(19)	CNN(20)	CNN(50)	0,878	0,112	0,009
MGI(100)	CNN(19)	CNN(20)	0,840	0,152	0,008
MGI(100)	MGI(70)	CNN(100)	0,783	0,212	0,005

5 lentelė. Daugumos balsavimo rezultatai. Skliausteliuose prie tinklo pavadinimo - epochų skaičius

Iš lentelės matome, kad panašaus pajėgumo tinklų daugumos konsensusas šiek tiek pagerina rezultatus (tačiau gana nežymiai). Tuo tarpu prastesnio tinklo įtraukimas į konsensumą su dviem stipresniais tinklais neturi ženklios įtakos. Stipraus tinklo įtraukimas į konsensumą su dviem silpnais tinklais pagerina rezultatą.

5.4.2. Balsavimas su svoriais

Balsavimo su svoriais eksperimentiniai rezultatai pademonstruoti lentelėje (žr. 7 lentelę).

Tinklas)	Tinklas	Tinklas	Tikslumas	Klaidingai teigiami	Klaidingai neigiami
CNN(19)	CNN(20)	CNN(50)	0,876	0,117	0,005
MGI(100)	CNN(19)	CNN(20)	0,863	0,128	0,009
MGI(100)	MGI(70)	CNN(100)	0,840	0,158	0,003

6 lentelė. Balsavimo su svoriais rezultatai. Skliausteliuose prie tinklo pavadinimo - epochų skaičius

Iš lentelės matome, kad panašaus pajėgumo tinklų svorinio konsensuso atveju gaunami šiek tiek geresni, tačiau labai panašūs rezultatai, kaip ir atskirų tinklų. Tuo tarpu prastesnio tinklo įdėjimas į konsensusą su geresniais tinklais visiškai nekenkia rezultatams - rezultatai yra net šiek tiek geresni, nei pavienių tinklų atveju. Gero tinklo įdėjimas tarp prastesnių tinklų smarkiai gelbėja situaciją - jungtinio tinklo rezultatas yra panašus į geriausio tinklo rezultata.

6. Sukurtų tinklų palyginimas ir įvertinimas

Visi trys sukurti neuroniniai tinklai gali spręsti plaučių auglių klasifikavimo uždavinį, tačiau tikslumas nėra labai aukštas. Tinklai yra labai skirtingi savo charakteristikomis - jos apžvelgiamos lentelėje (žr. lentelę).

Tinklas)	Treniravimo greitis	Tikslumas	Klaidingai teigiamų kiekis	Klaidingai neigiamų kiekis
CNN	vidutinis	vidutinis	vidutinis	žemas
MGI-CNN	vidutinis	žemas	didelis	žemas
VGG16-CNN	lėtas	vidutinis	vidutinis	žemas

7 lentelė. Balsavimo su svoriais rezultatai. Skliausteliuose prie tinklo pavadinimo - epochų skaičius

Labiausiai nuviliantis tinklas - MGI-CNN, kurio rezultatai pasirodė esantys visiškai priešingi, nei buvo daroma prielaida remiantis moksliniu straipsniu - klaidingai teigiamų kiekis šio tinklo klasifikavimo rezultatuose yra didelis.

Šio rezultato priežastimi gali būti dažnas nuotraukų matmenų keitimas interpoliacijos pagalba - tokiu būdu prarandama nemažai informacijos, kuri skiria tikrai teigiamą nuotrauką nuo panašios į ją.

Daugiausiai vilčių teikianti konvoliucinio tinklo architektūra - modifikuotas VGG16 konvoliucinio tinklo modelis. Vienintelis pastebėtas šio modelio trūkumas - itin ilgas treniravimo laikas. Tai gali iš dalies nulemti ir išplėstinių konvoliucijų naudojimas - kaip teigiama internetiniuose resursuose, „PyTorch“ karkase ši operacija yra veikianti gana lėtai.

Pozityviai vertintinas ir balsavimų mechanizmas - jis retai pablogina klasifikavimo rezultatus, o dažnu atveju atvirkščiai - juos pagerina. Nepaisant to, jį vartoti reiktų atsargiai - itin prasto tinklo įtraukimas į konsensuso balsavimą gali pabloginti rezultatą, ypač daugumos balsavimo atveju. Todėl perspektyviausias konsensuso mechanizmas - balsavimas su svoriais, kurių reikšmė priklauso nuo anksčiau tinklo parodytų tikslumo rezultatų.

Darbo rezultatai

Atlikus literatūros analizę ir eksperimentinę darbo dalį, pasiekti tokie rezultatai:

- Atlikta įvairių rūšių konvoliucinių neuroninių tinklų algoritmų analizė
- Praktiškai implementuoti 3 skirtingi konvoliuciniai neuroniniai tinklai
- Praktiškai sukurtas konsensuso mechanizmas bendram pastarųjų neuroninių tinklų klasifikavimo sprendimui priimti.
- Praktiškai ištirti ir palyginti implementuoti tinklai bei jų veikimas taikant konsensuso mechanizmą.

Išvados

Atlikus literatūros analizę ir eksperimentą buvo padarytos išvados:

- Moksliniuose straipsniuose pasiūlyti konvoliucinių neuroninių tinklų architektūros modeliai yra įdomūs, tačiau ne visados rodo geresnius rezultatus, nei plačiau žinomi konvoliucinių neuroninių tinklų modeliai.
- MGI-CNN tipo konvoliuciniai neuroniniai tinklai turi ne mažiau klaidingai teigiamų klasifikavimo rezultatų, nei kiti konvoliuciniai neuroniniai tinklai.
- VGG16 tipo konvoliuciniai neuroniniai tinklai yra tinkami spręsti plaučių auglių klasifikavimo užduočiai, tačiau yra labai lėtai treniruojami. Šio darbo autoriaus pasiūlytas VGG16 stiliaus konvoliucinis neuroninis tinklas nėra geresnis už tradicinio VGG16 stiliaus konvoliucinį neuroninį tinklą - bent jau esant tokiam nedideliame konvoliucinių filtrų skaičiui.
- Konsensuso mechanizmas pagerina klasifikavimo rezultatus. Svorinis balsavimas turi didesnę pozityvią įtaką klasifikavimo tikslumui, nei daugumos balsavimas.

Literatūra

- [19] Lung cancer statistics, 2019. <https://www.wcrf.org/dietandcancer/cancer-trends/lung-cancer-statistics>.
- [CZG⁺19] Ximin Cui, Ke Zheng, Lianru Gao, Dong Yang ir Jinchang Ren. Multiscale spatial-spectral convolutional network with image-based framework for hyperspectral image-classification. *Remote Sensing*, 11:2220, 2019-09. doi: 10.3390/rs11192220.
- [DDQ⁺17] Wei Dai, Chia Dai, Shuhui Qu, Juncheng Li ir Samarjit Das. Very deep convolutional neural networks for raw waveforms. P. 421–425, 2017-03. doi: 10.1109/ICASSP.2017.7952190.
- [GZT⁺] Lei Geng, Siqi Zhang, Jun Tong ir Zhitao Xiao. Lung segmentation method with dilated convolution based on vgg-16 network. *COMPUTER ASSISTED SURGERY*. doi: 10.1080/24699322.2019.1649071.
- [Hus] Muneeb ul Hussan.
- [YIS19] Muhamad Yani, S Irawan ir M.T. S.T. Application of transfer learning using convolutional neural network method for early detection of terry's nail. *Journal of Physics: Conference Series*, 1201:012052, 2019-05. doi: 10.1088/1742-6596/1201/1/012052.
- [KYC⁺19] Bum-Chae Kim, Jee Seok Yoon, Jun-Sik Choi ir Heung-Il Suk. Multi-scale gradual integration cnn for false positive reduction in pulmonary nodule detection. *NEURAL NETWORKS*, 115:1–10, 2019-07. issn: 0893-6080. doi: 10.1016/j.neunet.2019.03.003.
- [Kur19] Olga Kurasova. Skaitmeninio intelekto ir sprendimų priėmimo paskaitų medžiaga, 2019.
- [MDB17] Lukas Mosser, Olivier Dubrulle ir Martin Blunt. Stochastic reconstruction of an oolitic limestone by generative adversarial networks. *Transport in Porous Media*, 2017-12. doi: 10.1007/s11242-018-1039-9.
- [SZ14] Karen Simonyan ir Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*, 2014-09.
- [TCE20] Mesut Toğaçar, Zafer Cömert ir Burhan Ergen. Classification of brain mri using hyper column technique with convolutional neural network and feature selection method. *Expert Systems with Applications*:113274, 2020-02. doi: 10.1016/j.eswa.2020.113274.
- [ZCZ⁺19] Rui Zhang, Chao Cheng, Xuehua Zhao ir Xuechen Li. Multiscale Mask R-CNN-Based Lung Tumor Detection Using PET Imaging. *MOLECULAR IMAGING*, 18:1–8, 2019-JUL 30. issn: 1536-0121. doi: {10.1177/1536012119863531}.

Sąvokų apibrėžimai

- Dirbtinių neuroninių tinklų konsensusas - neuroninių tinklų klasifikavimo rezultatų apjungimas ir galutinio bendro rezultato nustatymas.
- Interpoliacija - duomenų aibės matmenų pakeitimas taip, kad duomenys neprarastų savo vientisumo. Pvz. nuotraukos mastelio keitimas.
- Mask-R-CNN - konvoliucinio neuroninio tinklo rūšis, kuri išplečia R-CNN konvoliucinio tinklo veikimą „kaukių“ taikymu.
- Tenzorius - duomenų rinkinys, kurio elementų tvarka ir atstumai vienas tarp kito yra svarbūs. Pirmos eilės tenzorius vadinamas vektoriumi, antros eilės - matrica.
- VGG16-CNN - konvoliucinio neuroninio tinklo rūšis, kuriai būdinga daug konvoliucinių sluoksnių.

Priedas nr. 1

Praktinės darbo dalies naudojimo instrukcijos

Pasiruošimo instrukcijos:

- 1) Nuklonuojam repozitoriją <https://github.com/maddddd/bakalauras.git>
- 2) Parsisiunčiam Anaconda v2020.2 ir ją suinstaliuojam.
- 3) (Nebūtina) Parsisiunčiam Nvidia Cuda v10.2 ir ją suinstaliuojam (tinka tik Nvidia grafinėms kortoms).
- 4) Atidarome "Anaconda prompt" konsolę ir sukuriame naują conda aplinką:

```
conda create --name bakalauras - bus sukurta nauja aplinka, vardu "bakalauras"
```

- 5) Aktyvuojame aplinką "Anaconda prompt" konsolėje:

```
activate bakalauras - bus aktyvuota aplinka, vardu "bakalauras"
```

- 6) Suinstaliuojame PyTorch:

```
conda install pytorch torchvision cudatoolkit=10.2 -c pytorch
```

Dauguma kitų reikalingų paketų, kaip pvz. numpy ar PIL yra suinstaliuojami kartu su PyTorch paketu arba yra iškart įtraukti į Anaconda distribuciją, todėl jų papildomai instaliuoti nereikia.

- 7a) Jei nenorime patys išgavinėti nuotraukų iš kompiuterinės tomografijos duomenų: Įeiname į katalogą "data", kuriame yra "pics.7z" failas ir jį ten pat išsarchyvuojame.

- 7b) Jei norime patys išgauti nuotraukas iš kompiuterinės tomografijos duomenų (gali užtrukti ilgai): "Anaconda prompt" konsolėje, aktyvavę 4 žingsnyje sukurtą aplinką, rašome:

```
pip install SimpleITK
```

- 8b) Tada parsisiunčiame kompiuterinės tomografijos duomenis iš:

```
https://luna16.grand-challenge.org/Download/
```

Reikalingi failai - subset0.zip : subset9.zip bei candidates_v2.csv failai
Candidates_v2.csv failą tiesiog įdedame į "data" katalogą, o subset0.zip : subset9.zip failus išsarchyvuojame "data" kataloge.

- 9b) Paleidžiame prep-pics.py skriptą, esantį "code\parse-pics\" kataloge ir laukiame, kol nuotraukos bus išgautos. Jei skriptas lūžta, reikia padidinti operacinės sistemos virtualią atmintį.

Naudojimo instrukcijos:

Norint naudoti tinklus, kviečiantysis kodas privalo patikrinti, ar jis yra pagrindinė programos gija, t.y. kodas turi būti apgaubtas konstrukcijos:

```
if __name__ == "__main__":
```

1) Norint naudoti neuroninį tinklą, pirma reikia jį sukurti ir ištreniruoti. Tai yra daroma tiesiai neuroninių tinklų failuose kataloge "code", pvz. parašome or_cnn.py failo gale:

```
cnn = cnn = CNN(0.001, 50, 32, 'train', 40)
```

- bus sukurtas tinklas su į atmintį užkrautais treniravimo duomenimis.

2) Treniruojame tinklą:

cnn.train() - treniravimo eiga aprašoma konsolėje. Tinklo elementų svoriai bus apmokyti.

3) Išsaugom tinklą (svarbu, jei norime jį vėliau testuoti):

```
tools.save_model(cnn, 'or_cnn') - tinklo svoriai bus išsaugoti "trained_nets" kataloge.
```

4) Įkeliamo tinklą iš naujo iš failo (prasminga, jei norime pakeisti duomenų aibę, pvz. norime dirbti su 'test' tipo duomenimis)

```
path = tools.get_model_path_in_hdd(cnn, 'or_cnn')  
cnn = tools.load_model(path, 'or_cnn', 'test', 50)
```

5) Testuojame tinklą - testavimas ne tik ištestuoja tinklą, bet ir įrašo testavimo rezultatų vidurkius į failą "params.txt", esantį kataloge "trained_nets".

Šie duomenys yra svarbūs, jei norime tinklą naudoti drauge su kitais tinklais (konsensusu).

Tinklų konsensuso testavimas:

Konsensuso tinklas pats savaime neegzistuoja, jis tiesiog naudoja kitų, jau ištreniruotų tinklų svorius ir testavimo nustatytų tikslumų rezultatus.

Norint testuoti tinklus drauge, reikia consensus.py failo gale, po apgaubiančia "if __name__ == "__main__":" konstrukcija, inicializuoti klasės Consensus objektą. Ši klasė priima neuroninių tinklų objektus ir jų tikslumus kaip konstruktoriaus parametrus.

Patogiausia šiuos parametrus gauti funkcijos load_networks_from_paths() pagalba:

```
nets, accs, pos, negs = load_networks_from_paths()
```

Metodas parašo, kurie tinklai diske yra tinkami pasirinkimui, kaip dalyviai.

Metodas yra interaktyvus, t.y. jis paprašo vartotojo įvesti norimų tinklų numerius.

Eiliškumas tas, pagal kurį jie buvo išvardinti anksčiau.

Numeruojama nuo nulio. Pasirinkimo konsolėje pavyzdys:

0 1 3 <Enter>- bus pasirinkti 3 tinklai iš išvardintų tinklų sąrašo.

Konsensuso objekto inicializavimo pavyzdys:

```
cons_net = Consensus(nets, accs, pos, negs, 'weighted', 0.001, 1, 32)
```

Testavimas naudojant konsensusą:

`cons_net.predict()` - bus atspausdinti testavimo rezultatai konsolėje.