



UNIVERZITET U NIŠU

ELEKTRONSKI FAKULTET



**PRIMENA KONVOLUCIJSKIH NEURONSKIH MREŽA ZA
KLASIFIKACIJU MEDICINSKIH SLIKA: PROBLEM
NEDOVOLJNE KOLIČINE PODATAKA**

MASTER RAD

Studijski Program: Računarstvo i informatika

Mentor:

Prof. dr Leonid Stoimenov

Kandidat:

Aleksandra Stanojević 918

Niš, decembar 2025. godine

Master rad

PRIMENA KONVOLUCIJSKIH NEURONSKIH MREŽA ZA KLASIFIKACIJU MEDICINSKIH SLIKA: PROBLEM NEDOVOLJNE KOLIČINE PODATAKA (predlog teme)

PRIMENA KONVOLUCIJSKIH NEURONSKIH MREŽA ZA KLASIFIKACIJU MEDICINSKIH SLIKA: PROBLEM KLASNE NERAVNOTEŽE (predlog teme)

ANALIZA EFIKASNOSTI KONVOLUCIJSKIH NEURONSKIH MREŽA ZA KLASIFIKACIJU MEDICINSKIH SLIKA PRI RADU SA OGRANIČENOM KOLIČINOM PODATAKA (još jedna moguća verzija)

Zadatak:

Proučiti....

Student:

Aleksandra Stanojević 918

Komisija za odbranu:

Predsednik

Datum prijave rada: _____ Član

Datum predaje rada: _____

Datum odbrane rada: _____ Član

PRIMENA KONVOLUCIJSKIH NEURONSKIH MREŽA ZA KLASIFIKACIJU MEDICINSKIH SLIKA: PROBLEM NEDOVOLJNE KOLIČINE PODATAKA

SAŽETAK

//TODO

Ključne reči: Konvolucijske neuronske mreže, Klasifikacija, Mali skup podataka, Nebalansirani skup podataka, COVID-19, bakterijska infekcija

CONVOLUTIONAL NEURAL NETWORKS APPLICATION FOR MEDICAL IMAGE
CLASSIFICATION: SMALL DATASET PROBLEM

ABSTRACT

//TODO

Key words: Convolutional Neural Network, Classification, Small dataset, Imbalanced dataset, COVID-19, bacterial pneumonia

Rad [21] -> Keywords: Coronavirus Pneumonia, COVID-19, Intelligent Medical System, X-ray Image Analysis, Deep Learning, Computer-Aided Diagnosis

Rad [36] -> Keywords: Deep learning, Convolutional neural networks, COVID-19, Coronavirus, Transfer learning, Deep feature extraction, Chest X-ray images

Sadržaj

Uvod	6
Problem nedovoljne količine podataka (nebalansirani skup podataka)	8
Povezani radovi	11
Duboko učenje	17
Tipovi neuronskih mreža	18
Kako funkcioniše duboko učenje?	18
Algoritam Bekpropagacije	20
Stohastički opadajući gradijent	20
Algoritamske varijacije Stohastičkog opadajućeg gradijenta	21
Lančanje izvoda funkcija	23
Primena dubokog učenja	24
Primena dubokog učenja u medicini - analiza medicinskih slika	24
Koraci u procesu klasifikacije medicinskih slika	26
Preprocesiranje	26
Izdvajanje karakteristika	27
Konvolucijske neuronske mreže	27
Sloj konvolucije (engl. Convolutional layer)	28
Sloj sažimanja (engl. Pooling layer)	29
Ravnjanje (engl. Flatten layer)	30
Klasifikacija	30
Funkcije aktivacije	31
Evaluacija	32
K-struka unakrsna validacija	33
Tehnike prevazilaženja problema nedovoljne količine podataka	34
Pretreniranje (engl. overfitting) i problem nedovoljne količine podataka	34
Transfer učenja	35
Fino podešavanje	37
Tehnike na nivou podataka	38
Augmentacija podataka	38
Naduzorkovanje podataka	38
Generisanje sintetičkih podataka korišćenjem GAN-a	39
Tehnike na nivou algoritma	41
Ponderisana kategorička unakrsna entropija	41

Fokalni gubitak	41
Hibridne tehnike	42
Korišćene tehnologije	43
WSL (Windows podsistem za Linux)	43
Visual Studio Code	43
Python	43
Anaconda (Miniconda)	43
TensorFlow	43
Keras	44
Implementacija	45
Specifične karakteristike oboljenja	45
Izbor vrste radiografskih snimaka	46
Korišćeni skupovi podataka	47
Učitavanje i preprocesiranje	48
Augmentacija podataka	51
Transfer učenja	52
Izdvajanje karakteristika	52
Fino podešavanje	53
Rezultati istraživanja	54
Zaključak	54
Literatura	55

Uvod

Jedan od najvećih problema koji se javlja prilikom rada na razvoju sistema za kompjuterski potpomognutu dijagnozu¹ (*engl. computer-aided diagnosis*) je nedovoljna količina podatka, a to se najviše odnosi na medicinske slike. Razvoj dobrog sistema za detekciju bolesti podrazumeva veliku količinu podataka visokog kvaliteta da bi mogao pravilno da nauči karakteristične osobine i obrazce koji su povezani sa određenom bolesti. Kada je na raspaganju samo mali skup podataka tada postoji rizik da sistem postane prespecijalizovan za ograničene podatke, a posledica je ta da će donositi pogrešne zaključke u radu sa do tada ne viđenim podacima.

U ovom radu će kao reprezentativni primer bolesti u vezi koje se jasno javlja nedovoljna količina podataka biti COVID²-19. Naime, u prvih par godina pandemije, količina dostupnih medicinskih slika sa potvrđenom infekcijom bila je jako ograničena, što se najbolje može primetiti kod najranijih istraživačkih radova na ovu temu. Međutim, sada kada je prošlo više vremena, količina dostupnih slika je mnogo veća, pa samim tim je možemo iskoristiti za validaciju efikasnosti konvolucijske neuronske mreže. Pri tome vršićemo i poređenje više različitih pristupa koji su najčešće proučavani u periodu pandemije, a za koje se tvrdilo da pomažu da se delimično ili čak potpuno prevaziđe problem ograničene količine podataka.

COVID-19 je virus koji se pojavio krajem 2019. godine u Kineskoj oblasti Wuhan, a u 2020. godini proširio na ceo svet i postao svetski zdravstveni problem [1-5], a vanredna situacija za javno zdravlje od međunarodne zabrinutosti za COVID-19 okončana je tek maja 2023. godine [8]. Glavna metoda koja se koristila za detekciju infekcije je lančana reakcija polimeraznom reverznom transkripcijom (skr. RT-PCR) koja može detektovati SARS³-CoV-2 RNA iz respiratornih uzoraka [9]. Međutim, ova metoda se pokazala kao dosta specifična, dugotrajna, naporan ručni postupak uz ponekad i pogrešne rezultate (lažni negativni i lažni pozitivni), a u periodu vrhunca pandemije javljaо se i problem nedostatka kompleta za testiranje usled velike potražnje. Ovo je inspirisalo mnogo istraživača da proučava mogućnosti automatske detekcije virusa korišćenjem rendgenskih slika pluća (*engl. Chest X-Ray - CXR*) ili slika pluća korišćenjem metode kompjuterizovane tomografije (*engl. CT scan*). Neki radovi su došli i do zaključka da dijagnostikovanje korišćenjem slika pluća daje bolje rezultate od RT-PCR metode [10]. Ovoj ideji doprinela je i činjenica da su medicinska istraživanja već i pre pokazala da je

¹ **Kompjuterski potpomognuta dijagnoza** je tehnologija koja koristi algoritme i tehnike mašinskog učenja kako bi pomogla lekarima i drugim medicinskim stručnjacima u donošenju kliničkih odluka. Služi kao komplementarno sredstvo i pomažu lekarima da postave tačnije i efikasnije dijagnoze.

² **COVID** je skraćenica za bolest korona virusa (*engl. Coronavirus Disease*).

³ **SARS** je skraćenica za teški akutni respiratorni sindrom (*engl. Severe Acute Respiratory Syndrome*).

infekcija pluća koju izaziva COVID-19 drugačija u odnosu na druge infekcije i da je na snimcima pluća radiolozi mogu detektovati [11-13].

Klasifikacija rendgenskih slika pluća, međutim, nije novi problem u polju veštačke inteligencije. Naime, konvolucijske neuronske mreže (*skr. CNN*) su jedan od najpopularnijih modela dubokog učenja koji se koristi za digitalnu obradu slike. Do prodora ove metode došlo je sa *ImageNet* takmičenjem [14], na kome je stopa greške za prepoznavanje objekata prepolovljena u odnosu na dotadašnje metode. U istraživanjima povezanim sa COVID-19 infekcijom akcenat je na korišćenju CNN-a, ili radi kreiranja metoda koji će najkorektnije detektovati segmente na snimcima pluća koji ukazuju na infekciju ili radi klasifikacije slika pluća na ona sa i bez infekcije i to sve, naravno, uz jako ograničen skup podataka.

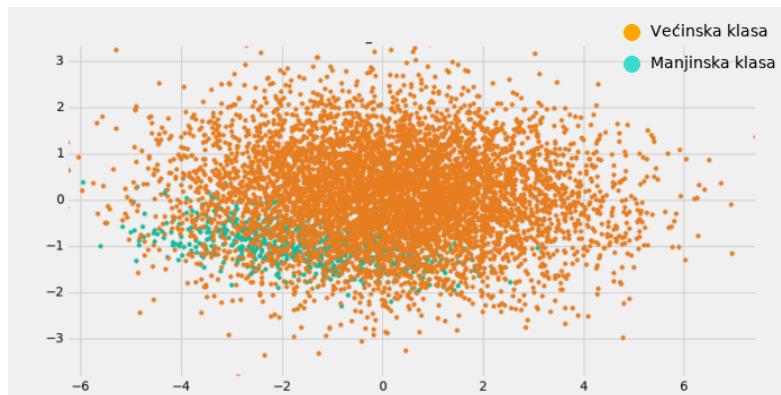
Ovaj rad se bavi proučavanjem digitalne obrade rendgenskih slika pluća korišćenjem dubokog učenja, odnosno konvolucijske neuronske mreže radi klasifikacije medicinskih slika u slučaju nedovoljne količine podataka. Najpre je dat osvrт na relevantne radove koji su se bavili klasifikacijom rendgenskih slika pluća sa i bez COVID-19 infekcije i pristupima koje su koristili za prevazilaženje problema nedovoljne količine podataka. Potom je predstavljeno duboko učenje kao metoda za digitalnu obradu slika i detaljno obrađen metod konvolucijske neuronske mreže i načina njenog rada od prihvatanja ulaznih podataka do izdavanja rezultata. Takođe, predstavljene su tehnike za prevazilaženje problema nedovoljne količine podataka, kao i tehnike transfera učenja kojima se izbegava preterano podudaranje sa podacima za treniranje ili pretreniranje (*engl. overfitting*).

Teorijski koncepti obrađeni u radu su zatim korišćeni prilikom realizacije praktičnog dela zadatka. Najpre su predstavljene korišćene tehnologije, a zatim i sama implementacija, od skupa podataka koji su korišćeni, preko preprocesiranja, kreiranja konvolucijske neuronske mreže do prikaza matrica, dijagrama i evaluacije korišćenjem standardne metrike. Korišćene tehnike za prevazilaženje problema nedovoljne količine podataka, odnosno slika sa potvrđenom COVID-19 infekcijom, su klasična augmentacija podataka, podešavanje težina, fokalni gubitak, klasično naduzorkovanje, SMOTE tip naduzorkovanje podataka i generativne suparničke mreže (*engl. Generative Adversarial Networks*). Uz svaku od njih korišćena je tehnika transfera učenja za par modela CNN-a koji su davali najbolje rezultate u prethodnim istraživanjima i k-struka unakrsna validacija, a zatim je vršeno poređenje performansi ovih različitih tehnika.

Problem nedovoljne količine podataka (nebalansirani skup podataka)

Neujednačene raspodele podataka prirodno se javljaju u mnogim primenama gde se klasa od interesa javlja sa smanjenom učestalošću, uključujući podatke koji se nalaze u dijagnozi bolesti, otkrivanju prevara, računarskoj bezbednosti i prepoznavanju slika. Unutrašnja neravnoteža je rezultat prirodnih učestalosti podataka, na primer medicinskih dijagnoza gde je većina pacijenata zdrava. Spoljašnja neravnoteža, sa druge strane, uvodi se kroz spoljne faktore, kao što su prikupljanje ili skadištenje podataka. [30]

Kada govorimo o medicinskim slikama, nebalansirani skupovi podataka i nedovoljna količina podataka su uobičajeni i često neizbežni izazovi. Određena stanja, pogotovo ako govorimo o retkim bolestima, se ne javljaju često u stvarnom svetu, što rezultuje u malom broju dostupnih podataka za manjinsku klasu koja je ujedno i klasa od interesa. Pored toga, prikupljanje medicinskih podataka ograničeno je zabrinutostima za privatnost, etičkim propisima i potrebom za stručnim anotacijama. Ovi faktori utiču na to da normalna i uobičajena stanja dominiraju, dok su klinički značajne abnormalnosti prisutne u mnogo manjoj meri.



Slika XZ. Vizualizacija raspodele klasa u prostoru karakteristika

Prethodno navedeni problemi su ujedno i razlozi zbog kojih je razvoj sistema za kompjuterski potpomognutu dijagnozu veoma težak zadatak, a jedan od glavnih je što istraživači moraju da prikupe potvrđene dijagnoze od strane stručnog lica. Na primer, kod radioloških snimaka, precizna lokacija i obim abnormalnosti moraju biti određeni od strane radiologa. Takva ručna anotacija od strane eksperta je dugotrajna i skupa. Najbolje anotacije nastaju kada više obučenih posmatrača označava istu abnormalnost, tako da se može dobiti probabilistička procena pouzdanosti lokacije lezije. Zbog toga mnogi sistemi za kompjuterski potpomognutu dijagnozu uključuju svega nekoliko desetina ili stotina potvrđenih slučajeva. Bolji dokazi obično dolaze iz

velikih skupova podataka, reda hiljada ili više, koji su nedostizni za sve osim za finansijski najjače studije. Na primer, NLST studija je obuhvatila preko 53.000 pacijenata i koštala više od 250 miliona dolara. Malo studija može da dostigne ove brojeve. [31]

U ovom radu govoriće se o problemu nedovolje količine podataka prilikom razvoja sistema za kompjuterski potpomognutu dijagnozu za klasifikaciju CXR slika sa potvrđenom COVID-19 infekcijom kao klasom od interesa. Naime, infekcija COVID-19 virusa idealan je primer bolesti kod koje je bilo nemoguće prikupljanje velike količine rendgenskih slika pluća sa potvrđenom infekcijom u prvim mesecima (godinama) od njegove pojave. Sa druge strane radiografski snimci pluća bez abnormalnosti, kao i snimci slika sa potvrđenom pneumonijom, bakterijskom i virusnom, bili su dostupni u mnogo većim količinama. Još jedan razlog zbog koga je baš ova tema jako zanimljiva, je za proučavanje efikasnosti konvolucijske neuronske mreže pri radu sa jako malom količinom podataka, jer su prvi realizovani radovi, objavljeni u periodu od 2020. do 2022. godine koristili veoma mali broj CXR slika sa potvrđenom COVID-19 infekcijom - od najmanje 25 do najviše 358. Ovi su radovi koristili transfer učenja za puno različitih modela CNN-a, a u narednom poglavlju biće dat detaljan osvrt na njih sa posebnim akcentom na pristupima koje su koristili za prevazilaženje problema nedovoljne količine podataka.

Pojedina istraživanja fokusirala su se na evaluaciju ovih radova, sa ciljem da se identifikuju postignuća, ograničenja i uočene greške. U radu [33] utvrđeno je da su dva modela, koja su tokom treniranja i testiranja na malim skupovima podataka pokazivala izuzetno dobre performanse, pokazala značajan pad tačnosti pri evaluaciji na do tada neviđenim podacima. Prema radu [34] model je sklon većem riziku od pristrasnosti kada je treniran na manje od 500 COVID-19 pozitivnih slika i manjem riziku kada se koristi 2000 slika.

Noviji radovi, poput [32], [40], [44], [48] i [53] zasnivaju se na znatno većim skupovima podataka, koji obuhvataju od 1345 do 4000 CXR slika sa potvrđenom COVID-19 infekcijom, pa stoga radovi realizovani i objavljeni u periodu 2023. do 2025. neće biti detaljnije obrađivani, jer ne koriste ekstremno male količine podataka kao oni koji im prethode. Međutim, važno je naglasiti da i ovi radovi sa dosta većom količinom podataka, koja ponovo nije u onom rangu u kome bi trebala biti za treniranje modela dubokog učenja, koriste metode za prevazilaženje problema nedovoljne količine podataka i naglašavaju važnost njegovog rešavanja i česte pojave u polju obrade medicinskih slika. Rad [32] se posvećuje detaljnjoj analizi transfera učenja i koristi augmentaciju podataka kao i [44] i [53], dok rad [40] predlaže malo modifikovanu verziju fokalnog gubitka, prilagođenu višeklasnoj klasifikaciji za prevazilaženje ovog problema. Autori rada [53] vrše analizu efikasnosti tri modela dubokog učenja InceptionV3, EfficientNetL i ConvNeXtXLarge pri radu sa malim (24, 48 i 100 slika), srednjim (248 i 500 slika) i većim (1000, 2000 i 4000 slika) skupom CXR slika pluća radi klasifikacije na četiri klase. Rezultati njihovog istraživanja pokazuju snažnu vezu između performansi modela dubokog učenja i

količine podataka koja se koristi za njihovo treniranje. Navode da je i malo povećanje broja slika sa 24 na 100 rezultovalo značajnom povećanju tačnosti. Na kraju rada diskutuju da je njihov model testiran na samo jednom skupu podataka te da samim tim može doći do drugačijih rezultata u slučaju rada modela sa do tada neviđenim podacima, što tvrde da će obraditi u svom narednom istraživanju.

Ideja ovog rada je da se utvrdi koliko su tehnike za prevazilaženje nedovoljne količine podataka zaista efikasne kada se radi sa ekstremno malim skupovima podataka. Radovi realizovani u prvim godinama od pojave COVID-19 virusa prijavljivali su jako dobre rezultate prepoznavanja koji neretko prelaze 90% tačnosti uz jako dobre stope senzitivnosti, međutim neki od njih su u kasnijim godinama osporeni, jer se nisu dobro pokazali prilikom prepoznavanja do tada neviđenih podataka. Stoga će u ovom radu konvolucijske neuronske mreže biti trenirane i validirane na jednom, jako malom skupu podataka uz upotrebu nekoloko tehnika za prevazilaženje nedovoljne količine podataka, koje su se pokazale kao najefikasnije, a zatim će biti testirane na drugom skupu podataka koji će biti značajno veći. Na taj način, moći ćemo jasno da uporedimo performanse mreža i utvrdimo da li i koje su tehnike za prevazilaženje problema nedovoljne količine podataka zaista imale učinka.

Iako je Svetska zdravstvena organizacija 5. maja 2023. godine zvanično proglašila kraj pandemije [8], istraživanja usmerena na detekciju i klasifikaciju CXR snimaka pluća sa i bez COVID-19 infekcije i dalje su aktuelna. Ova tema ostaje relevantna, pre svega zato što COVID-19 i dalje postoji u populaciji, ali i zbog potrebe da se izvuku pouzdani zaključci o primeni veštačke inteligencije u analizi medicinskih slika, što može biti od ključnog značaja u sličnim situacijama u budućnosti.

Povezani radovi

Analiza i detekcija COVID-19 infekcije su bile obimno istraživane teme u vreme trajanja pandemije, a veliki broj radova bavio se pitanjima vezanim za otkrivanje infekcije pomoću različitih pristupa dubokog učenja koristeći CXR i/ili CT slike pluća. Kako je tema ovog rada obrađena na primeru detekcije i klasifikacije virusa COVID-19 u nastavku će biti dat osvrt na postignute rezultate kao i korišćene načine za prevazilažene problema nedovoljne količine podataka i to koristeći skup podataka koji će biti korišćen i u praktičnom delu ovog rada.

Kao što je već prethodno rečeno klasifikacija rendgenskih slika pluća nije novi problem u polju dubokog učenja. Objavljeno je dosta skupova podataka i nad njima obučeno puno neuronskih mreža koje su dale jako dobre rezultate. Već sa početkom pandemije su Dr. Joseph Cohen i saradnici, istraživači sa Univerziteta Stanford objavili otvoreni skup podataka COVID-19 [17] koji sadrži CXR i CT slike pluća za koje je potvrđena infekcija ovim virusom. Od objavljuvanja ovog skupa, mnogi istraživači pokušali su da klasifikuju slike sakupljene kombinovanjem ovog i drugih skupova CXR slika, a neki od njih dati su u nastavku.

Narin i saradnici [18] su kreirali tri skupa podataka (covid i čista pluća/pluća sa bakterijskom infekcijom/virusnom infekcijom) nad kojima su vršili binarnu klasifikaciju. Predložili su CNN model zasnovan na transferu učenja korišćenjem ImageNet skupa podataka. Za prevazilaženje problema nedovoljne količine podataka koristili su augmentaciju podataka i 5-struku unakrsnu validaciju. Istraživanje je pokazalo superiornost ResNet50 modela u odnosu na pet korišćenih modela za sva tri skupa podataka i dobili tačnost od 96.1%, 99.5% i 99.7%.

Apostolopoulos i Mpesiana [19] su vršili klasifikaciju na tri klase i predložili CNN model zasnovan na transferu učenja korišćenjem ImageNet skupa podataka za pet modela. Za prevazilaženje problema nedovoljne količine podataka koristili su 10-struku unakrsnu validaciju. Kao najbolji model za klasifikaciju ovog specifičnog zadatka izdvojena je MobileNetV2 jer je pokazala najbolju stopu specifičnosti, te je validirana na skupu kome su dodate slike sa drugim virusnim infekcijama i tačnost iznosi 96.78% za dve i 94.72% za tri klase⁴.

Wang i saradnici [20] su predložili novu arhitekturu Covid-Net, trenirali su je na njihovom skupu podataka COVIDx, dok je pretreniranje vršeno na ImageNet skupu. COVIDx je u trenutku pisanja rada sadržao svega 358 slika sa potvrđenom covid infekcijom. Za prevazilaženje

⁴ Apostolopoulos i Mpesiana razlikuju dve vrste tačnosti, 3-klasna tačnost predstavlja ukupnu tačnost modela pri razlikovanju tri klase (čista pluća, pneumonija i covid19), dok 2-klasna tačnost meri samo sposobnost modela da razlikuje Covid-19 od ostalih stanja.

problema nedovoljne količine podataka koristili su klasičnu augmentaciju i pretreniranje. Dostignuta tačnost u ovom radu je 93.3%.

Hemdan i saradnici [21] su kreirali programski okvir za duboko učenje Covidx-Net i vršili binarnu klasifikaciju nad balansiranim skupom podataka od svega 50 slika pluća. Ovaj rad predstavlja uporednu studiju sedam arhitektura dubokog učenja. Za prevazilaženje problema nedovoljne količine podataka koristili su pretreniranje vršeno na ImageNet skupu i k-struku unakrsnu validaciju. Dostigli su tačnost od 90%, a ističu da treba raditi na usavršavanju modela MobileNetV2 zbog njegove brzine i mogućnosti korišćenja na pametnim telefonima.

Chakraborty i saradnici [35] predlažu svoju CNN Corona-Nidaan pogodnu za korišćenje na jednostavnijim uređajima kao što je mobilni telefon jer smanjuje računsku složenost i porede je sa pet arhitektura dubokog učenja. Za prevazilaženje problema nedovoljne količine podataka koriste naduzorkovanje, augmentaciju podataka i podešavanje težina. Utvrđeno je da, pored njihove, VGG19 ima najbolje performanse dostaigavši 93% tačnosti.

Rahimzadeh i Attar [36] su predstavili ideju o konkatenaciji modela Xception i ResNet50V2 što je moguće jer generišu matricu odlika istih dimenzija. Za prevazilaženje problema nedovoljne količine podataka koristili su augmentaciju i treniranje u osam faza. Za svaku fazu su slike čistih pluća i sa pneumonijom bile drugačije, a sve covid slike ponavljale. Tako je mreža mogla da nauči karakteristike manjinske klase, pa je prosečna dostaigunuta tačnost za prepoznavanje covid slika 99.5%, dok je opšta tačnost 91.4%.

Abbas i Abdelsamea [37] predlažu model DeTraC (decomposition-transfer-composition) i vrše klasifikaciju na tri klase. Dekompozicija podrazumeva podelu svake klase na k potklasa, a kompozicija ponovo sastavljanje radi kreiranja finalnih rezultata. Dekompozicija i augmentacija su tehnike korišćene za prevazilaženje problema nedovoljne količine podataka, kao i pretreniranje metodom plitkog i metodom dubokog podešavanja za više modela. Najveća dostaignuta tačnost je 97.35% za slučaj korišćenja dubokog metoda podešavanja VGG19 mreže unutar DeTraC modela.

Singh Punn i Agarwal [41] porede performanse više modela CNN-a treniranih na ImageNet skupu podataka. Takođe porede i učinak prevazilaženja problema nedovoljne količine podataka naduzorkovanjem (sa augmentacijom podataka) i podešavanjem težina. Pored prethodnog porede i binarnu sa višeklasnom klasifikacijom (za tri i četiri klase). Rezultati istraživanja pokazali su se najboljim za slučaj kada se koristi NASNetLarge model, naduzorkovanje podataka i vrši binarna klasifikacija. Postignuta tačnost za ovaj slučaj je 98%.

Hira i saradnici [42] poseban akcenat u svom radu stavljuju na transfer učenja metodom finog podešavanja i proučavaju njegovu efikasnost u slučaju klasifikacije CXR slika pluća. Formirali su četiri skupa podataka, a za prevazilaženje problema nedovoljne količine podataka koriste augmentaciju i 5-struku unakrsnu validaciju. Najbolje performanse pokazuju Se-ResNeXt-50, tačnost za binarnu klasifikaciju je 99.32%, a za višeklasnu 97.55%.

Abdulkadir Karaci [43] predlaže kaskadni model VGGCOV19-NET koji predstavlja VGG19 model kome su izmenjeni potpuno povezani slojevi i čiji su ulazni podaci, izlazni podaci iz YOLOv3 algoritma (metod za detekciju regija od značaja). Istiće značaj transfera učenja usled nebalansiranog skupa podataka i nedovoljne količine podataka. Koristi 5-struku unakrsnu validaciju i računa težinski prosek za agregaciju (sabranu ocenu) preciznosti i odziva preko svih klasa. Postignuta tačnost za binarnu klasifikaciju je 99.84%, a za višeklasnu 97.16%.

Zebin i saradnici [45] porede tri modela trenirana na ImageNet skupu podataka i vrše troklasnu klasifikaciju balansiranog skupa CXR slika. Takođe, vrše poređenje efikasnosti augmentacije podataka i generativne suparničke mreže (*skr: GAN*). Uz pomoć GAN-a od slika čistih pluća generišu 100 covid slika koje uvrštavaju u trening podatke. Utvrđili su, korišćenjem VGG16 mreže, da drugi pristup daje bolje rezultate, pa ga primenjuju na ostalim modelima. Koriste Gran-CAM metodu za vizualizaciju regija slike koje su uticale na predikcije. Najbolje performanse dao je EfficientNetB0 model sa tačnošću od 96.8%.

Paul i saradnici [46] predlažu metod za kombinovanje izlaznih predikcija više različitih modela, treniranih na ImageNet podacima. Predloženi metod dodeljuje određene težine izlazu svakog klasifikatora, a konačno predviđanje vrši izvođenjem ponderisanog proseka⁵ tih izlaza. Njihov metod nazvan Grupisanje obrnute zvonaste krive (*engl. Inverted Bell Curve Ensemble*), dodelu težina bazira na nivou samopouzdanja predikcije odgovarajućeg klasifikatora. Tačnost ovog sistema je 99.66% na prvom i 99.84% na drugom skupu podataka.

Zhang i saradnici [47] koriste ResNet18 model treniran na ImageNet skupu podataka kao deo svog modela čiji izlazi se respektivno propagiraju na klasifikator i sistem za detekciju anomalija. Za prevazilaženja problema nedovoljne količine podataka koriste klasičnu augmentaciju i sistem za detekciju anomalija pomoću dubokog učenja⁶. Cilj predloženog modela je detekcija COVID-19 anomalija sa visokim stepenom osetljivosti (*engl. sensitivity*) i on dostiže stopu osetljivosti od 90%, a stopu specifičnosti od 87.84%.

⁵ Ponderisani prosek je vrsta proseka koja daje različite težine različitim vrednostima na kojima se izračunava.

⁶ **Sistem za detekciju anomalija pomoću dubokog učenja** podrazumeva treniranje modela da prepoznae COVID-19 snimke kao "anomalije" u odnosu na normalne (ne-COVID-19) slike, što pomaže modelu da bolje generalizuje i prepozna COVID-19 čak i sa malo primera.

Ozturk i saradnici [49] su konstruisali svoj model nazvan DarkCovidNet po ugledu na DarkNet19 model koji je pokazao dobre rezultate kod YOLO algoritma, sistema za detekciju objekata, čija je osnova. Njihov model sastoji se od sedamnaest konvolucijskih slojeva i koriste normalizaciju serija-a i Leaky ReLU funkciju aktivacije. Vršili su binarnu klasifikaciju i dostigli tačnost od 98.8% i klasifikaciju na tri klase gde su dostigli tačnost od 87.02%.

Pored radova koji su koristili slike iz Cohen-ovog skupa ima puno i onih koji su kreirali sopstveni skup snimaka i koji je ostao zatvoren za javnost, te im se ne može pristupiti, pa stoga ovde neće biti obradivani. Tabela XX predstavlja poređenje opisanih modela koji datiraju iz perioda od 2020. do 2022. godine, sa početka pandemije.

Studija	Skup podataka	Transfer učenja	Tehnike prevazilaženja problema nedovoljne količine podataka	Tačnost
Narin i saradnici [18]	341 covid19 2800 čista pluća 2772 bakt. inf. 1493 dr. virusi	ResNet50 , ResNet101, ResNet152, InceptionV3, Inception-ResNetV2	1. Augmentacija 2. Transfer učenja 3. 5-struka unakrsna validacija	96.1% (covid i čista pluća) 99.5% (covid i virusna inf.) 99.7% (covid i bakt. inf.)
Apostolopoulos i Mpesiana [19]	Skup1: 224 covid19 700 bakt. inf. 504 čista pluća Skup2: 224 covid19 714 pneumonia 504 čista pluća	VGG19 , MobileNetV2 , Inception, Xception, InceptionResNetV2	1. Transfer učenja 2. 10-struka unakrsna validacija	97.4% (Skup1, 2 klase) 92.85% (Skup1, 3 klase) 96.78% (Skup2, 2 klase) 94.72% (Skup2, 3 klase)
Wang i saradnici [20]	358 covid19 5538 pneumonia 8066 čista pluća	ImageNet - pretreniranje COVIDx - treniranje (Covid-Net, VGG19, ResNet50)	1. Transfer učenja 2. Augmentacija podataka	93.3% (covid, čista pluća i pneumonia) 90.6% (covid, čista pluća i pneumonia)
Hemdan i saradnici [21]	25 covid19 25 čista pluća	VGG19 , DenseNet201 , InceptionV3, ResNetV2, InceptionResNetV2, Xception, MobileNetV2	1. Transfer učenja 2. Balansirani skup podataka 3. k-struka unakrsna validacija	90% (covid i čista pluća) 90% (covid i čista pluća)
Chakrabarty i saradnici [35]	245 covid19 5551 pneumonia 8066 čista pluća	Corona-Nidaan , VGG19 , MobileNetV2, InceptionResNetV2, InceptionV3, DenseNet201	1. Transfer učenja 2. Naduzorkovanje 3. Augmentacija 4. Podešavanje težina	95% (covid, čista pluća i pneumonia) 93% (covid, čista pluća i pneumonia)

Rahimzadeh i Attar [36]	180 covid19 6054 pneumonia 8851 čista pluća	ImageNet skup (konkatenirane Xception i ResNet50V2)	1. Transfer učenja 2. Augmentacija 3. Treniranje u 8 faza	91.4% (covid, čista pluća i pneumonia)
Abbas i Abdelsamea [37]	105 covid19 11 SARS 80 čista pluća	DeTraC (VGG19, AlexNet, ResNet, GoogleNet, SqueezeNet)	1. Transfer učenja (duboko/plitko podešavanje) 2. Augmentacija 3. Dekompozicija klasa	97.35% (covid, sars i čista pluća)
Singh Punn i Agarwal [41]	108 covid19 515 pneumonia 58 tuberkuloza 533 čista pluća	ResNet, InceptionV2, InceptionResNetV2, DenseNet169, NASNetLarge	1. Transfer učenja 2. Podešavanje težina 3. Naduzorkovanje sa augmentacijom 4. 4-struka unakrsna validacija	98% (covid i non-covid (čista pluća, tuberkuloza, bakterijske i druge virusne infekcije))
Hira i saradnici [42]	224 covid19 400 bakt. inf. 314 dr. virus 504 čista pluća Poseban skup (2358 bakt. inf, 1345 virusna 1346 čista pluća)	AlexNet, GoogleNet, ResNet-50, Se-ResNet-50, DenseNet121, Inception V4, Inception ResNet V2, ResNeXt-50 i Se-ResNeXt-50	1. Transfer učenja (fino podešavanje) 2. Augmentacija 3. 5-struka unakrsna validacija	99.32% (covid i čista pluća) 97.55% (covid, čista pluća i bakterijska inf.) 96.89% (covid, čista pluća bakt. inf. i virusna inf.) 95.98% (poseban skup)
Abdulkadir Karaci [43]	125 covid19 500 pneumonia 500 čista pluća	VGG19, Cascade VGG19, VGGCOV19-NET, CascadeVGGCOV19-NET	1. Transfer učenja 2. 5-struka unakrsna validacija	99.84% (covid i čista pluća) 97.16% (covid, čista pluća i pneumonia)
Zebin i saradnici [45]	202 covid19 300 pneumonia 300 čista pluća	VGG16, ResNet50, EfficientNetB0	1. Transfer učenja 2. Balansirani skup podataka 3. Augmentacija 4. Generisanje slika korišćenjem GAN-a 5. 5-struka unakrsna validacija	96.8% (covid, čista pluća i pneumonia)
Paul i saradnici [46]	Skup1: 219 covid19 1345 dr. virus 1341 čista pluća Skup2: 563 covid19 283 ne-covid19	DenseNet161, ResNet18, VGG16, AlexNet, InceptionV3 i Predloženi model⁷	1. Transfer učenja 2. Augmentacija (rotacija slika) 3. Kombinovanje rezultata više CNN-a	99.66% (Skup1) 99.84% (Skup2)

⁷ Kombinovanje rezultata tri modela (DenseNet161, ResNet18 i VGG16) korišćenjem grupisanja obrnute zvonaste krive (*engl. Inverted Bell Curve Ensemble*).

Zhang i saradnici [47]	100 covid19 1431 pneumonia	ResNet18	1. Transfer učenja 2. Augmentacija 3. Sistem za detekciju anomalija	stopa osetljivosti - 90% stopa specifičnosti - 87.84%
Ozturk i saradnici [49]	127 covid19 500 pneumonia 500 čista pluća	DarkCovidNet	1. 5-struka unakrsna validacija 2. Balansirani skup podataka 3. CNN arhitektura male složenosti	98.08% (covid i čista) 87.02% (covid, čista pluća i pneumonia)

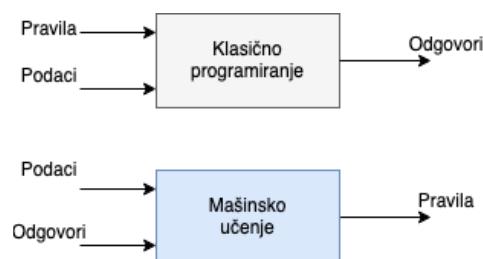
Tabela XX: Poređenje opisanih modela dubokog učenja koji za klasifikaciju koriste CXR slike iz Cohen-ovog skupa podataka COVID-19

Duboko učenje

Veštačka inteligencija nastala je još 1950-ih godina kada se prvi put javilo pitanje automatizacije intelektualnih zadataka. Ona predstavlja širu oblast koja obuhvata mašinsko učenje i duboko učenje. Prvi programi bili su za igranje šaha i podrazumevali su praćenje pravila, a ovaj pristup poznat je kao *simbolička veštačka inteligencija*. Međutim, kako on nije mogao da reši složenije logičke zadatke, razvio se novi pristup, poznat kao mašinsko učenje. [22]



SlikaXX: Veštačka inteligencija, mašinsko i duboko učenje



SlikaXX: Mašinsko učenje kao novi pristup u računarstvu

Mašinsko učenje počinje svoj razvoj sa pitanjem koje je postavio pionir veštačke inteligencije Alan Tjuring, a to je sposobnost maštine da samostalno uči kako da izvrši određen zadatak. U mašinskom učenju čovek dostavlja maštini podatke i odgovore koje očekuje, a ona sama definiše pravila na osnovu kojih će od podataka dobiti odgovore. Ono što je ideja ovog pristupa je pronalaženje transformacije koju je potrebno izvršiti nad podacima tako da se dobije odgovarajuće rešenje (odgovor). [22]

Jedan od načina podele algoritama za mašinsko učenje je na algoritme za:

- Nadgledano učenje (*engl. Supervised learning*) - algoritmi koji grade matematički model iz skupa podataka koji sadrže i ulaze i željene izlaze;
- Polunadgledano učenje (*engl. Semi-supervised learning*) - algoritmi koji razvijaju matematičke modele iz nepotpunih podataka, gde ulaznim podacima nedostaju labele;
- Nenadgledano učenje (*engl. Unsupervised learning*) - algoritmi koji razvijaju matematičke modele iz skupova podataka koji sadrže samo ulaze;

Duboko učenje je specifična podoblast mašinskog učenja koja podrazumeva pristup učenju kroz slojeve odnosno sukcesivne transformacije podataka. Dubina modela predstavlja koliko slojeva čini posmatrani model. Neki modeli uključuju desetine ili čak stotine uzastopnih slojeva, a učenje je automatsko prilikom izlaganja podacima za treniranje. Kod dubokog učenja, ovi slojeviti modeli su poznati pod nazivom **neuronske mreže**.

Tipovi neuronskih mreža

Neuronske mreže su modeli mašinskog učenja koji odražavaju način na koji funkcioniše ljudski mozak. Osnovne jedinice neuronske mreže su neuroni, bazirani na biološkim neuronima, dok se veze između njih razvijaju kroz vreme, odnosno treniranje. Od pojave ovog modela, a naročito u poslednjoj deceniji radilo se na razvoju različitih algoritama dubokog učenja što je rezultiralo kompleksnijim arhitekturama neuronskih mreža čije se strukture i parametri prilagođavaju specifičnim oblastima primene. Najznačajnije četiri arhitekture mreža su [7]:

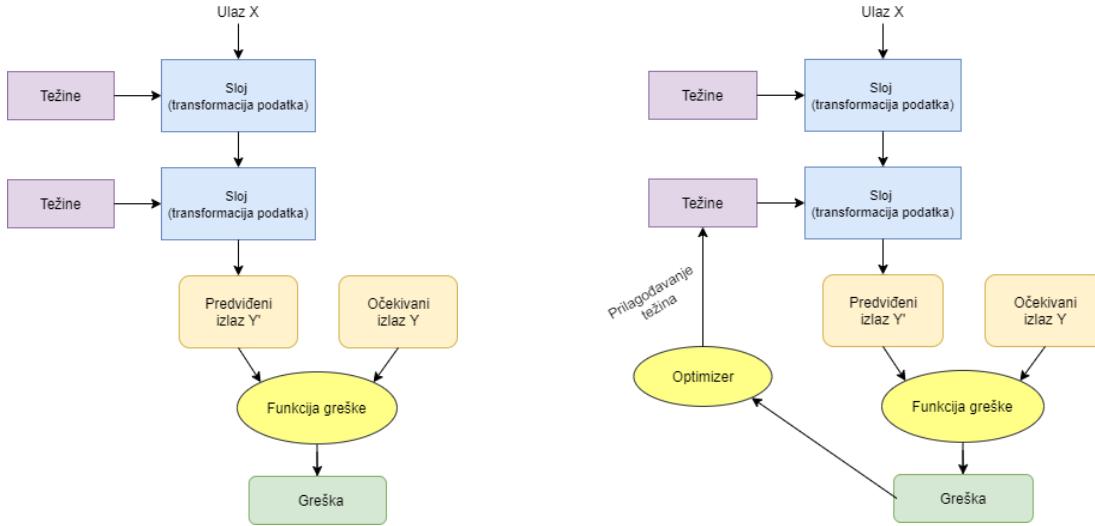
- **Nenadgledane pretrenirane mreže** (*engl. Unsupervised Pretrained Networks*) obuhvataju tri specifične arhitekture - Autoenkodere (*engl. Autoencoders*), Duboke mreže poverenja (*engl. Deep Belief Networks*) i Generativne suparničke mreže (*engl. Generative Adversarial Networks*):
 - **Generativne suparničke mreže** mogu da nauče uzorke u podacima i na osnovu njih izgenerišu nove, dosad neviđene podatke;
- **Konvolucijske neuronske mreže** (*engl. Convolutional Neural Networks*) imaju topologiju nalik rešetki i najčešće se koriste za obradu slika i detekciju objekata;
- **Rekurentne neuronske mreže** (*engl. Recurrent Neural Networks*) su specijalizovane za obradu sekvenčijalnih podataka. Od klasičnih neuronskih mreža se razlikuju po prisustvu povratne petlje u potpuno povezanim slojevima, koju koriste da uče iz sekvenci;
 - **Neuronske mreže za dugo-kratkoročnu memoriju** (*engl. Long Short Term Memory Networks*) imaju mogućnost učenja i pamćenja dugoročne zavisnosti;
- **Rekurzivne neuronske mreže** (*engl. Recursive Neural Networks*) imaju strukturu stabla i mogu naučiti hijerarhijske modele. Uključuju rekurzivnu primenu istog skupa težina na različite strukture tipa stabla.

Kako funkcioniše duboko učenje?

Način na koji duboko učenje funkcioniše podrazumeva da se specifikacija onoga što sloj radi sa ulaznim podacima čuva u njegovim težinama ili parametrima sloja. U ovom kontekstu, učenje znači pronalaženje skupa vrednosti za težine svih slojeva u mreži koji će omogućiti pravilno mapiranje ulaznih podataka u odgovarajuće izlazne podatke. Međutim, da bi se moglo upravljati izlazom neuronske mreže, neophodno je izmeriti koliko je on daleko od očekivanog rezultata, a to je zadatak funkcije greške (*engl. loss function*). [22]

Funkcija greške uzima predviđanja koja su izlazne vrednosti mreže i stvarne ciljeve (izlazne vrednosti koje želimo da dobijemo) i izračunava grešku. (Slika XX 1.8) Osnovni štos u dubokom učenju je upotreba ove greške kao vodilje za prilagođavanje težina u takvom smeru da će to

izazvati smanjenje greške (Slika XX 1.9). Ovo prilagođavanje težina je posao **optimizera** (*engl. optimizer*) koji implementira **Algoritam Bekpropagacije**, centralni algoritam u dubokom učenju.



Slika XX1.8: Funkcija greške meri kvalitet predviđanja neuronske mreže

Slika XX1.9: Optimizer koristi grešku da bi prilagodio težine

Razvijeno je dosta različitih funkcija greške, a koja će se koristiti zavisi od konkretnog problema nad kojim je treba primeniti. Neke od najpopularnijih su: [7]

Binarna unakrsna entropija (*engl. Binary Cross Entropy*) se koristi kod binarne klasifikacije, kada izlazne vrednosti mogu biti samo 0 ili 1. Ukoliko pretpostavimo da smo generisali model koji predviđa verovatnoću za y kada imamo dato x. Označimo taj model sa $f(x, \theta)$ gde θ predstavlja parametre (težine) modela. Tada možemo da je predstavimo formulom: [6]

$$-\sum_{i=1}^n y_i \log f(x_i, \theta) + (1 - y_i) \log(1 - f(x_i, \theta))$$

Kategorička unakrsna entropija (*engl. Categorical Cross Entropy*) se koristi u zadacima za klasifikaciju na više klase (višeklasna klasifikacija), a njena formula je: [6] $-\sum_{i=1}^n y_i \log f(x_i, \theta)$

Srednja kvadratna greška (*engl. Mean Squared Error*) koristi se kada je reč o problemima regresije. Računa se kao prosečna vrednost kvadrata razlike između vrednosti koje je model predvideo i očekivanih vrednosti. Formula za srednju kvadratnu grešku je: [7]

$$\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

Srednja apsolutna greška (*engl. Mean Absolute Error*) je varijacija srednje kvadratne greške, a njena vrednost je prosečna vrednost apsolutnih grešaka na celom skupu podataka: [7]

$$\frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$$

Srednja absolutna procentualna greška (*engl. Mean Absolute Percentage Error*) meri veličinu greške u procentima. Predstavlja prosek apsolutnih procenata grešaka za svaki ulaz: [7]

$$\frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

Hindž gubitak (*engl. Hinge Loss*) je najčešće korišćena funkcija greške kada mreža mora biti optimizovana koristeći metodu potpornih vektora (*engl Support Vector Machine*). Matematički se može predstaviti funkcijom: [7]

$$\frac{1}{n} \sum_{i=1}^n \max(0, 1 - y_i * \hat{y}_i)^2$$

Algoritam Bekpropagacije

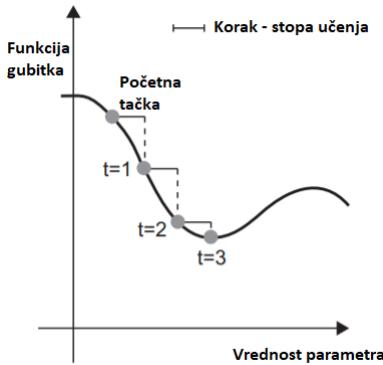
Najteži zadatak dubokog učenja je prilagođavanje težina modela (neuronske mreže) u takvom smeru da će to izazvati smanjenje greške. Međutim, postavlja se pitanje kako odrediti za pojedinačnu težinu u neuronskoj mreži da li je potrebno smanjiti je ili povećati i za koliko. Odgovor na ovo pitanje leži u činjenici da su sve operacije koje se koriste u neuronskoj mreži diferencijabilne, te se može izračunati gradijent na osnovu težina mreže. [22]

Stohastički opadajući gradijent

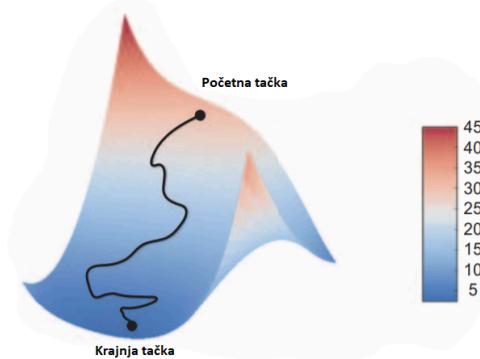
Ukoliko radimo sa diferencijabilnom funkcijom teoretski je moguće pronaći njen minimum, a to je tačka u kojoj je njen izvod jednak nuli. Potrebno je pronaći sve tačke u kojoj je izvod jednak nuli i zatim utvrditi za koju od njih funkcija ima najmanju vrednost. Međutim, ukoliko želimo da pronađemo minimum neuronske mreže, potrebno je da odredimo vrednosti za njene težine koje daju najmanju moguću vrednost za funkciju greške. Kod neuronskih mreža se za to koristi algoritam u četiri koraka koji malo po malo modifikuje parametre mreže na osnovu trenutne vrednosti funkcije greške: [22]

1. Određivanje x skupa podataka za obuku mreže i y skupa podataka koji predstavlja očekivane rezultate (ciljeve);
2. Treniranje neuronske mreže na x skupu podataka i dobijanje y_pred skupa podataka koji predstavljaju predviđene rezultate;
3. Izračunavanje greške, odnosno razlike između y i y_pred skupova podataka;
4. Izračunavanje gradijenta funkcije greške na osnovu parametara mreže;
5. Pomeranje parametara suprotno od gradijenta ($w -= step * gradient$), što izaziva smanjenje greške.

Opisani algoritam je poznat kao **stohastički opadajući gradijent mini-serije podataka** (engl. **Mini-batch Gradient Descent**). Važno napomenuti da ovaj algoritam iz ulaznog skupa podataka uzima u svakom koraku po jedan deo, te tako svaka iteracija radi sa novim podacima. Slika XX ilustruje šta se dešava kada mreža ima jedan parametar i kada imamo jedan uzorak za treniranje (jednodimenzionalni prostor). Kada govorimo o neuronskim mrežama, međutim, govorimo o **višedimenzionalnom prostoru**: svaka težina u neuronskoj mreži predstavlja novu dimenziju. Na Slika XX data je vizuelizacija algoritma u dvodimenzionalnom prostoru, da bi se mogla stvoriti neka slika o samom konceptu.



Slika XX: Stohastički opadajući gradijent mini serije u jednodimenzionalnom prostoru(jedan uzorak za treniranje)

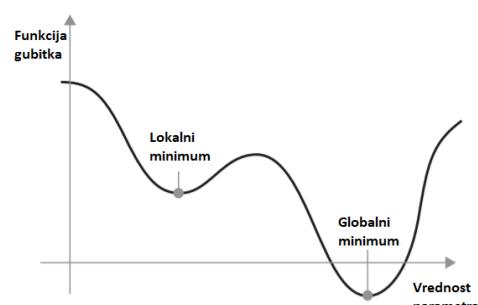


Slika XX: Stohastički opadajući gradijent mini serije u dvodimenzionalnom prostoru(jedan uzorak)

Algoritamske varijacije Stohastičkog opadajućeg gradijenta

Postoji više vrsta stohastičkog opadajućeg gradijenta, a razlikuju se po tome koliko uzimaju u obzir prethodna ažuriranja težina prilikom izračunavanja sledećeg. Postoje varijante sa impulsom koje se bave problemima brzine konvergencije i lokalnog minimuma. [22] Funkcije najčešće imaju više minimuma, a onaj koji je najbolje pronaći je globalni.

Na osnovu Slika XX se može zaključiti da je potrebno pažljivo odabratи korak, odnosno stopu učenja, jer ukoliko bude premali, biće potrebno mnogo iteracija da se dođe do minimuma ili se možemo zaglaviti kod lokalnog minimuma. Sa druge strane, ako je preveliki svaka iteracija nas može odvesti do kompletno slučajne lokacije na krivoj funkcije. [23] Rešenje ovog problema je impuls, a može se zamisliti kao spuštanje lopte niz krivu - ukoliko je impuls dovoljno jak loptica će proći lokalni minimum i završiti u globalnom. U praksi, to znači da ažuriramo parametar mreže i na osnovu prethodnog ažuriranja vrednosti gradijenta.



Slika XX: Lokalni i globalni minimum

Nesterov impuls

Ovaj algoritam predložio je Yurii Nesterov 1983. godine i njegova osnovna ideja je merenje gradijenta ne trenutnog nego onog blago unared u pravcu impulsa. Uzmimo da $\nabla 1$ predstavlja gradijent funkcije greške merene na početnoj tački t, a $\nabla 2$ gradijent u tački koja se nalazi na t + bm. Nakon nekog vremena, ova mala poboljšanja se sabiraju i Nesterov impuls je znatno brži od obične impuls optimizacije. Štaviše, treba imati na umu da kada impuls gura tegove preko doline, $\nabla 1$ nastavlja da gura dalje preko doline, dok $\nabla 2$ gura nazad prema dnu doline. Ovo pomaže u smanjenju oscilacija i time se brže konvergira. [23]

AdaGrad

AdaGrad algoritam prilagođava korak za svaki parametar. Ukoliko parametar funkcije greške označimo sa x_i i neka g_i^s označava gradijent za x_i u koraku s i uzmememo jednačinu $G = (g_i^0)^2 + (g_i^1)^2 + \dots + (g_i^{s-1})^2$, koja predstavlja sumu kvadrata gradijenata za svaki korak sve do prethodnog, onda pravilo ažuriranja algoritma glasi: $x_i = x_i - \frac{\alpha}{G^{\frac{1}{2}}} g_i^s$. Parametar α je globalna stopa učenja, koja se prilagođava za svaki parametar na osnovu prethodnih gradijenta. [6]

AdaGrad optimizer se ne preporučuje za obuku dubokih neuronskih mreža, jer nažalost često stane prerano pre nego što dostigne globalni minimum, ali ipak može biti efikasan za jednostavnije zadatke kao što je na primer linearna regresija. [23]

RMSProp

RMSProp algoritam prevazilazi slabost AdaGrad algoritma koji prebrzo usporava, zaustavlja se i ne dolazi do globalnog minimuma. Rešenje leži u tome što RMSProp sabira samo gradijente iz najnovijih iteracija, a ne sve gradijente od početka treniranja. Dakle sada se G računa samo za prethodnih W koraka, kao zbir eksponencijalno opadajućeg proseka kvadrata gradijenata. Pravilo ažuriranja RMSProp algoritma glasi:

$$x_i = x_i - \frac{\alpha}{RMS[g_i]} g_i^s \quad RMS[g_i] = \sqrt{E[(g_i^s)^2]} = \rho E[(g_i^s)^2]^{s-1} + (1 - \rho)(g_i^s)^2$$

Parametar ρ predstavlja eksponencijalno propadanje i za njegovu vrednost se uglavnom uzima 0.9 koja u najvećem broju slučajeva ima dobar performans i ne mora se podešavati.

Ovaj optimizer skoro uvek radi mnogo bolje od AdaGrad-a i klasičnog Nesterovog impulsa, a bio je preferirani algoritam mnogih istraživača sve dok se nije pojavio Adam optimizer. [6]

Adam optimizer

Adam je skraćenica za prilagodljivu procenu impulsa (*engl. Adaptive moment estimation*) i kombinuje klasični algoritam impulsa i RMSProp. Kao i klasični algoritam impulsa prati eksponencijalno opadajući prosek prethodnih gradijenata, a kao RMSProp prati eksponencijalno opadajući prosek kvadrata prethodnih gradijenata. [6]

$$\text{Pravilo ažuriranja Adam algoritma je: } x_i = x_i - \frac{\alpha}{E[(g_i^{s-1})]^2} E[g_i^{s-1}].$$

Kod ovog algoritma imamo dva parametra koji predstavljaju eksponencijalno propadanje, jedan za $E[g_i^{s-1}] = \frac{E[g_i^{s-1}]}{1-\rho_1}$ i drugi za $E[(g_i^{s-1})^2] = \frac{E[(g_i^{s-1})^2]}{1-\rho_2}$, ρ_1 i ρ_2 respektivno. [23]

Lančanje izvoda funkcija

U prethodnim algoritmima smo prepostavili da pošto je funkcija diferencijabilna, možemo izračunati njen izvod. Međutim, u slučaju neuronskih mreža govorimo o mnogo ulančanih operacija sa tenzorima, od kojih svaka ima poznati izvod. Na primer, imamo mrežu sa tri operacije sa tenzorima, a, b i c i matricama težina W1, W2 i W3:

$$f(W1, W2, W3) = a(W1, b(W2, c(W3)))$$

Matematika nas uči da se za takav lanac funkcija može pronaći izvod ukoliko koristimo sledeći identitet, poznat kao izvod složene funkcije: $f(g(x)) = f'(g(x)) * g'(x)$. Primena izvoda složene funkcije na izračunavanje gradijenta neuronske mreže predstavlja Algoritam Bekpropagacije (poznat i kao automatsko diferenciranje⁸ u obrnutom režimu). Bekpropagacija započinje finalnom greškom, zatim se kreće od unutrašnjih (dubljih) ka spoljašnjim slojevima i primenjuje izvod složene funkcije radi izračunavanja doprinosa koji je svaki parametar imao za vrednost greške. [22]

U današnje vreme i u budućnosti, ljudi će neuronske mreže kreirati koristeći specijalizovane biblioteke, kao što je TensorFlow, o kome će kasnije biti više reči, koje omogućavaju izvršenje čitavog algoritma Bekpropagacije pozivom samo jedne funkcije. Samim tim, neće nikada biti potrebno implementirati ovaj algoritam ručno.

⁸ Diferenciranje je postupak nalaženja izvoda u matematici.

Primena dubokog učenja

Iako je duboko učenje poprilično stara podoblast mašinskog učenja, njen uspon je krenuo tek početkom 2010-ih godina. Za samo nekoliko godina ova podoblast dostigla je izvanredne rezultate na perceptivnim problemima poput gledanja i slušanja - problema koji uključuju veštine koje su intuitivne za čoveka, ali su dugo bile nedostizne za mašine. Poseban proboj je duboko učenje imalo u istorijski teškim oblastima mašinskog učenja kao što su klasifikacija slika, prepoznavanje govora, konverzija teksta u govor i obrnuto, primena za razvoj digitalnih asistenata (Google Now, Siri, i dr.) i u skorije vreme mogućnost davanja odgovora na pitanja na prirodnom jeziku (ChatGPT, DeepSeek, Gemini, Claude, i dr.).

Još uvek se istražuje šta duboko učenje u potpunosti može da uradi. Ova oblast primenjena je na širok spektar problema, a neke od najvažnijih su one u oblasti nauke, farmacije i medicine.

Primena dubokog učenja u medicini - analiza medicinskih slika

Mašinsko učenje, duboko učenje i konvolucijske neuronske mreže napravile su značajan napredak u analizi medicinskih slika, uključujući analizu složenih radiografskih snimaka. Ove tehnike su pokazale veliki potencijal u izdvajajući značajnih obrazaca i karakteristika iz medicinskih slika koje možda nisu lako očigledne ljudskim posmatračima, vodeći zdravstvene radnike do tačnijih i objektivnijih dijagnoza i smanjenja dijagnostičkih grešaka. One takođe mogu pomoći zdravstvenim radnicima tako što ističu regije od interesa, pružaju kvantitativna merenja i predlažu potencijalne dijagnoze na osnovu naučenih obrazaca. Stoga, njihovo korišćenje u analizi medicinskih slika može poboljšati efikasnost dijagnostičkog procesa, omogućavajući brže tumačenje i donošenje odluka od strane zdravstvenih radnika.

Danas se poseban naglasak stavlja na dijagnostiku na mestu zbrinjavanja (POC dijagnostika), što podrazumeva pružanje medicinske nege i dijagnostičkog testiranja direktno ili blizu mesta gde se pacijent leči. Prenosivi radiografski aparati ili ručni uređaji mogu se koristiti za dobijanje slika, eliminujući potrebu za prevozom pacijenata do centralizovanih radioloških odeljenja. Ovaj pristup ima za cilj da pruži hitne zdravstvene usluge, omogućavajući pristup snimanju, jer smanjuje vreme između dobijanja slike i njene interpretacije. Ovo je posebno vredno u kritičnim situacijama ili hitnim slučajevima gde je brza dijagnoza ključna za pokretanje odgovarajućih dijagnostičkih odluka. Takva nega može poboljšati efikasnost zdravstvenih sistema i posebno je korisna u okruženjima sa ograničenim resursima ili udaljenim područjima. [40]

Tehnike dubokog učenja za obradu medicinskih slika su uspešno primenjene i to za veliki spektar problema, a neki od njih su [24]:

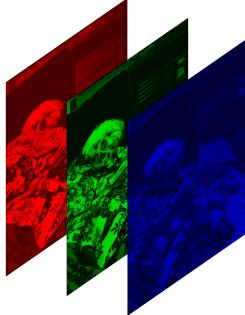
- Detekcija dijabetične retinopatije (izlečenje je povezano sa ranim otkrivanjem oboljenja);
- Otkrivanje histoloških i mikroskopskih elemenata (nivo ćelije i tkiva) što uključuje detekciju:
 - kožnih bolesti kao što su karcinom skvamoznih ćelija i melanoma;
 - karcinoma želuca i metaplazija želudačnog epitela;
 - karcinoma dojke;
 - malarije;
 - prisustva crevnih parazita;
 - tuberkuloze.
- Detekcija bolesti probavnog trakta, što podrazumeva detekciju upalnih procesa, krvarenja, infekcija i kancera u gastrointestinalnom traktu kao što su:
 - čirevi - izazivaju krvarenje u gornjem delu gastrointestinalnog trakta;
 - polipi, rak i divertikulitis - izazivaju krvarenje iz debelog creva;
 - celijakija, Kronova bolest, maligni i benigni tumor (bolesti tankog creva);
 - crevna opstrukcija;
 - čir na dvanaestopalačnom crevu;
 - sindrom iritabilnog creva;
 - krvarenje usled abnormalnih krvnih sudova (arteriovenske malformacije - angiodisplazije ili angioektazije).
- Detekcija bolesti srca, gde se posebno ističe automatsko izračunavanje kalcijumskog skora, koji je jedan od najboljih pokazitelja rizika od srčanog udara i ateroskleroze;
- Otkrivanje ćelija tumora u bilo kom delu čovekovog tela, što uključuje detekciju i benignih i malignih promena;
- Otkrivanje bolesti mozga - Alchajmerove i Parkinsonove bolesti.

Koraci u procesu klasifikacije medicinskih slika

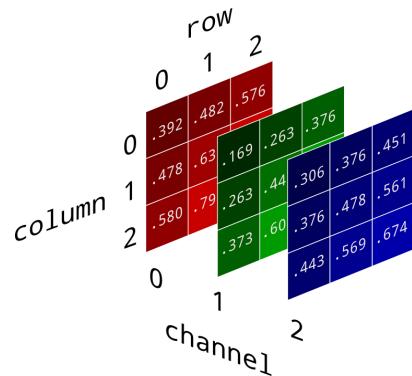
Ono što je najvažnije u slučaju zdravstvenog sektora je da sistem za dijagnostikovanje treba da da što tačnije rezultate, a za to je neophodno proći kroz osnove korake digitalne obrade medicinskih slika. Digitalni koraci obrade medicinskih slika su preprocesiranje, izdvajanje karakteristika, klasifikacija i evaluacija.

Preprocesiranje

Medicinske slike mogu biti pod uticajem smetnji odnosno izložene takozvanoj “buci” (*engl. noise*), koja se može javiti tokom prenosa slika. Prilikom preprocesiranja ove smetnje se smanjuju, a neželjeni podaci ručno odsecaju i prerađuju. Preprocesiranje se odvija kroz sledeće korake: pročišćavanje podataka (*engl. data cleaning*), integracija podataka, smanjenje skupa podataka (*engl. data reduction*) i transformacija skupa podataka. [25]



SlikaXY. Slika prikazana kroz 3 kanala (RGB)



SlikaXY. Predstavljanje slike preko 3 2D matrice

Slika je predstavljena 2D matricom u kojoj pikseli uzimaju vrednost iz opsega [0-255]. Slike koje imaju jedan kanal su slike na sivoj skali (*engl. grayscale*) i one su predstavljene jednom matricom piksela (kod standardnih slika postoje 3 kanala, po jedan za crvenu, zelenu i plavu boju) što ih čini pogodnim za obradu. Da bi se slika konvertovala u sivu, potrebno je prosečnu vrednost svakog piksela za svaki kanal smestiti na odgovarajuće mesto u novoj matrici.

Podaci koji se koriste kao ulazni moraju biti formatirani na odgovarajući način pre pohranjivanja neuronske mreže. Učitavanje podataka iz fajl sistema i dekodiranje JPEG slika u RGB mreže piksela su prvi koraci. Sledeći bitan korak je skaliranje piksela sa vrednosti iz opsega [0-255] na opseg [0-1]. Biblioteke mašinskog učenja uglavnom sadrže module koji služe kao pomoćni alati pri preprocesiranju slika i prethodno pomenuti koraci se mogu automatizovati.

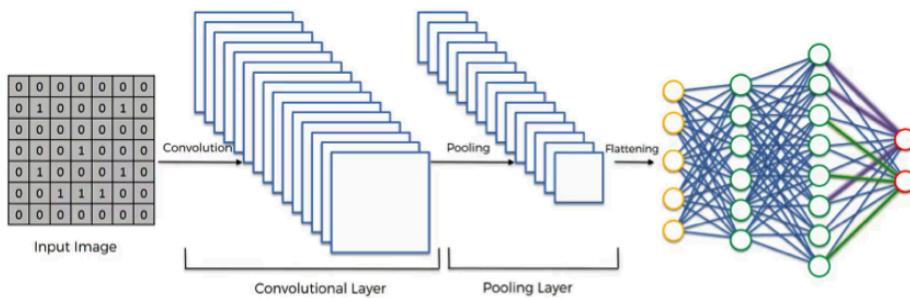
Izdvajanje karakteristika

Izdvajanje karakteristika je proces smanjenja dimenzionalnosti kojim se početni skup podataka smanjuje na grupe koje se mogu dalje obrađivati. Odlika velikih skupova podataka je veliki broj varijabli za čiju obradu je potrebno mnogo resursa. Izdvajanje karakteristika je naziv za metode koje biraju i/ili kombinuju varijable u karakteristike, efikasno smanjujući količinu podataka za obradu, a istovremeno tačno i u potpunosti opisujući izvorni skup podataka.

Nakon preprocesiranja ulaznih podataka oni se ulivaju u konvolucijsku neuronsku mrežu koja vrši izdvajanje karakteristika i smanjuje redundantnost. Način na koji konvolucijska neuronska mreža vrši izdvajanje karakteristika predstavljen je u nastavku ovog poglavљa.

Konvolucijske neuronske mreže

Do nedavno računari nisu mogli da izvrše neke trivijalne zadatke poput prepoznavanja objekta na slici ili prepoznavanja prirodnog govora. Razlog je taj što se kod čoveka percepcija odvija izvan područja svesti, unutar specijalizovanih slušnih, vizuelnih i drugih modula u mozgu. Međutim, zahvaljujući David H. Hubel-u i Torsten Wiesel-u koji su izveli serije eksperimenata na mačkama [26-27] i majmunima [28] i dali ključni uvid u strukturu vizuelnog korteksa (autori su dobili Nobelovu nagradu za fiziologiju ili medicinu 1981. godine), rodila se ideja koja je vremenom evoluirala do danas široko poznatih Konvolucijskih neuronskih mreža.



Slika XX: Gradivni blokovi konvolucijske neuronske mreže

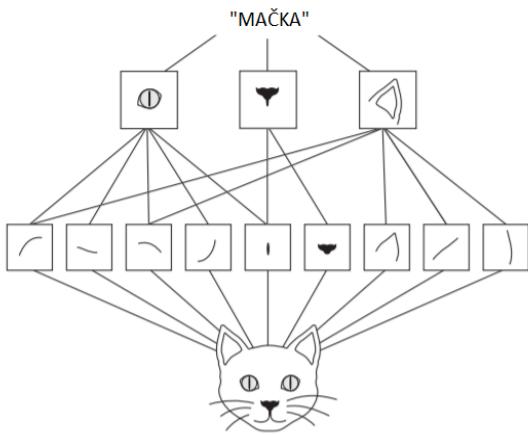
U poslednjih nekoliko godina, zahvaljujući povećanju računarske snage, količine podataka za treniranje i novim pronašćima za problem treniranja neuronskih mreža, konvolucijske neuronske mreže su uspele da postignu nadljudske performanse na velikom broju kompleksnih vizuelnih zadataka. Danas one predstavljaju model dubokog učenja koji se gotovo univerzalno koristi za probleme računarskog vida, a njeni gradivni blokovi prikazani su na Slika XX. [23]

Neuronska mreža predstavlja sistem koji se sastoji od određenog broja međusobno povezanih procesnih elemenata ili čvorova koje nazivamo veštačkim **neuronima**. Arhitektura neuronske mreže predstavlja specifično povezivanje neurona u jednu celinu, a njena struktura se razlikuje po broju slojeva. Konvolucijska neuronska mreža sastoji se od ulaznog i izlaznog sloja, kao i više skrivenih slojeva. Skriveni slojevi se obično sastoje od niza konvolucionih slojeva, slojeva sažimanja i potpuno povezanih slojeva.

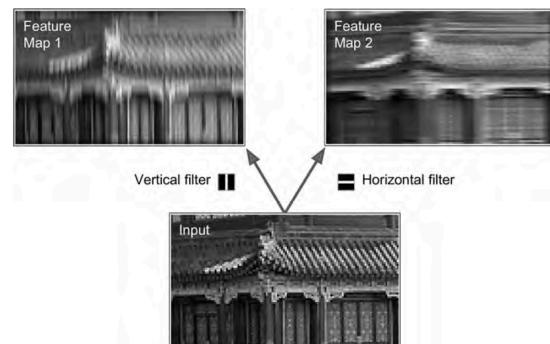
Sloj konvolucije (engl. Convolutional layer)

Najvažniji gradivni blok konvolucijske neuronske mreže je sloj konvolucije. Prvi skriveni sloj mreže uči lokalne obrasce, odnosno koncentriše se na najmanje karakteristike (elemente) sa slike koja se obrađuje (ulazni podatak), a zatim u narednim skrivenim slojevima ih kombinuje u veće karakteristike (elemente) i tako na dalje. Ova ključna odlika daje konvolucijskim neuronskim mrežama dve zanimljive osobine [22]:

- **Naučeni obrasci su nezavisni u odnosu na svoj položaj:** Nakon što nauči određeni obrazac u donjem desnom uglu slike, konvolucijska neuronska mreža ga može prepoznati bilo gde (na primer, u gornjem levom uglu). To čini konvolucijske neuronske mreže efikasnim pri obradi slika (vizuelni svet je u osnovi nezavistan u odnosu na svoj položaj);
- **Mogu naučiti prostorne hijerarhije obrazaca** (videti Slika XX): Prvi sloj konvolucije naučiće male lokalne obrasce kao što su ivice, a drugi sloj veće obrasce napravljene od karakteristika prvih slojeva itd. Tako mreža efikasno uči sve složenije i apstraktnije vizuelne koncepte (jer je vizuelni svet u osnovi prostorno hijerarhijski).



Slika XX: Prostorna hijerarhija: lokalni obrasci (oči, uši) se kombinuju u veće obrasce (mačka)



Slika XX: Upotreba dva različita filtera radi dobijanja dve različite mape karakteristika

Operacija konvolucije radi sa trodimenzionalnim tenzorima, takozvanim **mapama karakteristika**, sa dve prostorne ose (visina i širina), kao i dubinskom osom (osom kanala). Za RGB sliku, dimenzija dubinske ose je 3, a za crno-belu sliku dubina je 1. Operacija konvolucije

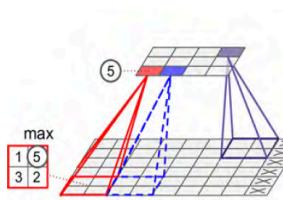
izvlači parčice iz ulazne mape karakteristika i na njima primenjuje istu transformaciju, stvarajući izlaznu mapu karakteristika. Ova mapa izlaznih karakteristika i dalje je trodimenzionalni tenzor, a njegova dubina može biti proizvoljna, jer dubina više ne predstavlja boje već **filtere**. [22]

Slika XX prikazuje dva moguća skupa težina, koji se nazivaju filteri. Prvi je predstavljen kao crni kvadrat sa vertikalnom belom linijom u sredini - neuroni koji koriste ove težine ignorisale sve u svom receptivnom polju, osim centralne vertikalne linije. Drugi filter je crni kvadrat sa vodoravnom belom linijom u sredini, pa će neuroni ignorisati sve, osim centralne horizontalne linije. Ukoliko svi neuroni u sloju koriste isti filter vertikalne linije, sloj će dati gornji levi izlaz na Slika XX. [23] Vertikalne bele linije se poboljšavaju dok se ostatak zamče. Slično tome, slika u gornjem desnom uglu se dobija, ako svi neuroni koriste filter horizontalne linije - vodoravne bele linije poboljšavaju dok su ostale zamčene. Dakle, sloj neurona koji koriste isti filter daje mapu karakteristika, koja ističe područja na slici koja su najsličnija filteru.

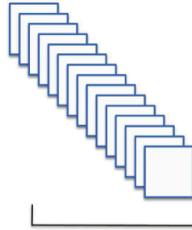
Tokom treninga, CNN pronalazi najkorisnije filtere za svoj zadatak i uči da ih kombinuje u složenije obrasce.

Sloj sažimanja (engl. Pooling layer)

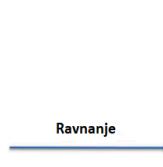
Sledeći sloj u konvolucijskoj neuronskoj mreži, nakon sloja konvolucije je sloj sažimanja. Cilj ovog sloja je da smanji ulaznu sliku radi smanjenja računarskog opterećenja, upotrebe memorije i broja parametara. Baš kao i u konvucionim slojevima, svaki neuron u sloju sažimanja povezan je sa ograničenim brojem neurona iz prethodnog sloja (receptivno polje). Međutim, neuroni iz ovog sloja nemaju težinu, jer sve što sloj radi je da agregira ulaze koristeći funkciju agregacije kao što je maksimum ili srednja vrednost. Na Slika XX prikazan je sloj za sažimanje koji koristi maksimum, što je i najčešće korišćena funkcija agregacije. Dakle, samo maksimalna ulazna vrednost u svakom neuronu prelazi na sledeći sloj, a ostali ulazi su odbačeni.



Slika XX: Sloj sažimanja koji koristi maksimum



Slika XX: Sloj ravnanja



Ulagi sloj za klasičnu veštačku neuronsku mrežu

Ovo je vrlo destruktivna vrsta sloja čak i sa sičušnim 2×2 parčićima i korakom 2, jer će izlaz biti dva puta manji u oba smera (tako da će njegova površina biti četiri puta manja). Odnosno biće odbačeno 75% ulaznih vrednosti. Sloj sažimanja obično radi na svakom ulaznom kanalu nezavisno, tako da je izlazna dubina jednaka ulaznoj dubini.

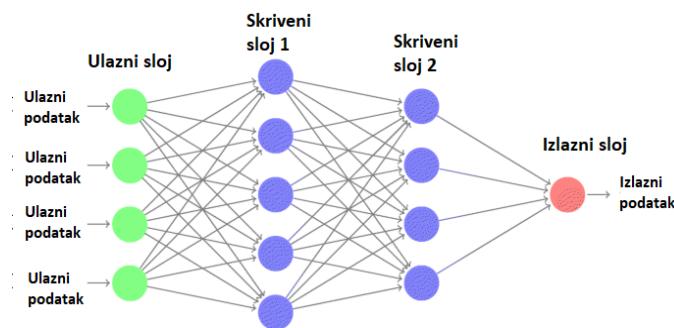
Ravnjanje (engl. Flatten layer)

Nakon završetka prethodna dva koraka imamo kao rezultat sažetu mapu karakteristika. Kao što i sam naziv ovog koraka glasi, sledeće što je potrebno uraditi je izravnati našu sažetu mapu karakteristika u jednu kolonu kao što je prikazano na Slika XX. Ono što se dešava nakon sloja ravnjanja je da na kraju dobijamo jako dugačak vektor ulaznih podataka koji zatim prolaze kroz veštačku neuronsku mrežu da bi se dalje obrađivali.

Klasifikacija

Najčešći zadaci mašinskog učenja su regresija (predviđanje vrednosti) i klasifikacija (predviđanje klase). Klasifikacija podrazumeva određivanje klase kojoj pripadaju elementi iz skupa podataka koji se obrađuje, odnosno predviđa klasu za ulaznu promenljivu. Najjednostavniji vid klasifikacije u mašinskom učenju je **binarna klasifikacija**, kod koje postoje samo dve moguće klase za ulazne podatke, dok postoji i **klasifikacija na veći broj klasa** koja predstavlja složeniji problem, pogotovo ako govorimo o velikom broju klasa.

U slučaju konvolucijske neuronske mreže klasifikacioni sloj dolazi nakon sloja konvolucije, sloja sažimanja i ravnjanja. Ovaj sloj sakuplja krajnje rezultate procesa konvolucije i sažimanja kao svoje ulazne podatke i za njih vrši klasifikaciju. Klasifikacioni sloj se sastoji od jednog ulaznog sloja, jednog ili više skrivenih slojeva i jednog izlaznog sloja (Slika XX). Svaki sloj osim izlaznog je u potpunosti je povezan sa sledećim slojem.

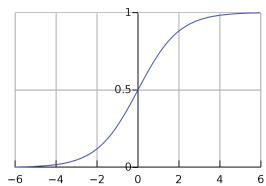


Slika XX: Klasifikacioni sloj konvolucijske neuronske mreže (potpuno povezani slojevi)

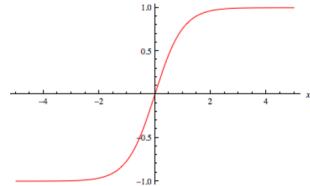
Matematičke jednačine koje određuju izlaznu vrednost neuronske mreže zovu se funkcije aktivacije. Funkcija aktivacije je povezana sa svakim neuronom u mreži i određuje da li on treba da se okine (aktivira) ili ne, na osnovu relevantnosti ulaznog podatka za predikciju modela.

Funkcije aktivacije

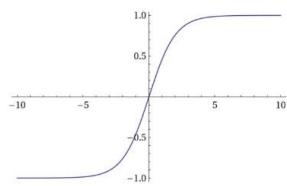
Klasična neurnoska mreža je samo linearni klasifikator, te kao takva ne može da rešava mnoge realne probleme, jer se oni uglavnom ne mogu predstaviti nekom linearom funkcijom. Da bi neuronske mreže mogle da nauče mnogo bogatije skupove transformacija, potrebno je uvesti nelinearnost u njihov dizajn, a tome služe takozvane funkcije aktivacije. Funkcije aktivacije pomažu pri normalizaciji izlaznih podataka iz svakog neurona na opseg od 0 do 1 ili između -1 i 1. Definicije i grafici nekih od najčešće korišćenih funkcija aktivacije date su u nastavku: [6]



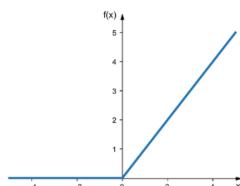
Slika XX: Grafik sigmoid funkcije



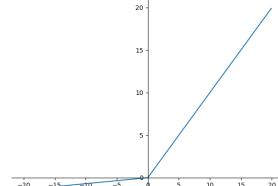
Slika XX: Grafik hiperbolične tangente



Slika XX: Grafik softmax funkcije



Slika XX: Grafik funkcije ispravljača



Slika XX: Grafik funkcije propuštajućeg ispravljača

Sigmoid funkcija se koristi kod zadataka sa binarnom klasifikacijom, odnosno kada izlazni sloj klasifikatora ima samo jedan neuron. Ona transformiše ulazne podatke koristeći: $f(x) = \frac{1}{1+e^{-x}}$

Hiperbolična tangenta se takođe koristi u slučaju binarne klasifikacije, a za razliku od sigmoid po većem opsegu vrednosti izlaznih podataka iz neurona (opseg -1 do 1). $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

Softmax funkcija za izlazni sloj daje distribuciju verovatnoće po svim klasama, te se koristi kod problema sa modeliranjem više klase odnosno kod višeklasne klasifikacije. $s(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$

Funkcija ispravljača (*engl. Rectified Linear Unit - ReLU*) se najčešće koristi u dubokom učenju zbog svoje jednostavnosti, a najviše zato što je u praksi pokazala jako dobre rezultate. Ona ulaz direktno prosleđuje na izlaz ako je veći od 0, a ukoliko je manji prosleđuje 0. $f(x) = \max(0, x)$

Funkcija propuštajućeg ispravljača (*engl. Leaky ReLU*) je varijacija ReLU operacije koja se koristi za sprečavanje odumiranja neurona. Za razliku od ReLU i Sigmoid funkcija aktivacije, koje imaju nultu vrednost u negativnom delu svojih derivata, ova funkcija ima malu epsilon vrednost za prevazilaženje problema odumiranja neurona. [49] $f(x) = \max(0.1 * x, x)$

Evaluacija

Kako bi se pratila uspešnost algoritma koji se koristi pri analizi i obradi slika potrebno je meriti performanse sistema. Evaluacija je neophodan korak kako bi se rezultat pratio i po potrebi sistem modifikovao i unapredio.

Matrica konfuzije je čest prikaz rezultata koji opisuju performanse modela mašinskog učenja i ona ima četiri moguća ishoda. Tačno pozitivan (*engl. True Positive*) je broj slika koje pripadaju određenoj klasi za koje je tačno predviđeno da toj klasi pripadaju. Tačno negativan (*engl. True Negative*) je broj slika koje ne pripadaju određenoj klasi za koje je tačno predviđeno da ne pripadaju toj klasi. Lažno pozitivan (*engl. False Positive*) je broj slika koje ne pripadaju određenoj klasi za koje je netačno predviđeno da pripadaju toj klasi. Lažno negativan (*engl. False Negative*) je broj slika koje pripadaju određenoj klasi za koje je netačno predviđeno da ne pripadaju toj klasi. [21]

	Predviđeno pozitivan	Predviđeno negativan
Stvarno pozitivan	Tačno pozitivan (TP)	Lažno negativan (FN)
Stvarno negativan	Lažno pozitivan (FP)	Tačno negativan (TN)

Tabela XX: Matrica konfuzije

Na osnovu vrednosti mogućih ishoda matrice konfuzije mogu se izračunati tipične mere performansi sistema za obradu slika kao što su stopa tačnosti, stopa preciznosti, stopa opoziva (stepen osetljivosti), stopa specifičnosti i F1-skor.

Stopa tačnosti je indikator ukupnog broja tačnih predviđanja modela, ali da bi ova mera dobro funkcionalisala potrebno je da imamo balansirani skup podataka. Ukoliko to nije slučaj, da bismo utvrdili tačne performanse modela, moramo da koristimo pored ove i druge metrike.

$$\text{Stopa tačnosti}(\%) = \frac{TP+TN}{TP+FP+FN+TN} \cdot 100\%$$

Stopa preciznosti se definiše kroz odnos broja tačnih predviđanja prema ukupnom broju ulaznih uzoraka, dok **stopa opoziva** (stepen osetljivosti) predstavlja udeo pozitivnih uzoraka slika koji su ispravno identifikovani kao pozitivni uzorci, a **stopa specifičnosti** predstavlja udeo negativnih uzoraka slika koji su ispravno identifikovani kao negativni uzorci. [46]

$$\text{Stopa preciznosti} = \frac{TP}{TP+FP} \quad \text{Stopa opoziva} = \frac{TP}{TP+FN} \quad \text{Stopa specifičnosti} = \frac{TN}{TN+FP}$$

F1-skor je ukupna mera tačnosti modela koja predstavlja harmonijsku sredinu stope preciznosti i stope opoziva i definise se na sledeći način:

$$F1\text{-skor} = 2 * \frac{Stopa\ preciznosti * Stopa\ opoziva}{Stopa\ preciznosti + Stopa\ opoziva}$$

U slučaju višeklasne klasifikacije važna metrika je i **logaritamski gubitak** i koristi se za smanjenje lažnih klasifikacija, jer meri sigurnost modela pri predviđanju klase.

$$\text{Logaritamski gubitak} = -\frac{1}{n} \sum_{i=1}^n \log PV$$

PV - predviđena verovatnoća za istinitu klasu
n - broj uzoraka

Odnos stope senzitivnosti i stope specifičnosti se može grafički prikazati kao **ROC kriva** (*engl. Receiver Operating Characteristic curve*). Savršeni model bi imao 100% senzitivnosti (za sve slike koje pripadaju određenoj klasi predviđeno je da pripadaju toj klasi) i 100% specifičnosti (za sve slike koje ne pripadaju određenoj klasi predviđeno je da ne pripadaju toj klasi).

Površina ispod ROC krive - **AUC** (*engl. Area Under the Curve*) pruža metriku za procenu performansi klasifikatora. Vrednost AUC leži u opsegu [0,5, 1], gde vrednost 0,5 označava performanse slučajnog klasifikatora, a vrednost 1 predstavlja savršeni klasifikator. Dakle, što je veća AUC, to su bolje performanse klasifikatora. [46]

K-struka unakrsna validacija

Evaluacija uz korišćenje test podataka za utvrđivanje mera performansi modela je k-cross validacija. Ova procedura bira uzorke za evaluaciju modela mašinskog učenja i ima samo jedan parametar - k, koji predstavlja broj grupa u koje je podeljen dati skup podataka. Vrednost k parametra se mora pažljivo odabratи na osnovu vrste podataka i veličine skupa podataka. Najčešće se uzima vrednost između 5 i 10, a ukoliko postoji dilema obično je dobra praksa opredeliti se za k = 10. Generalna procedura k-cross validacije opisana je u nastavku:

1. Izmešati skup podataka;
2. Podeliti skup podataka u k grupa;
3. Za svaku jedinstvenu grupu:
 - a. Uzeti grupu kao test skup podataka;
 - b. Uzeti preostale grupe kao trening skup podataka;
 - c. Trenirati model korišćenjem trening skupa podataka;
 - d. Evaluirati model korišćenjem test skupa podataka;
 - e. Zapamtitи rezultat evaluacije;
4. Izvesti zaključke na osnovu evaluacije;

Tehnike prevazilaženja problema nedovoljne količine podataka

Problem nedovoljne količine podataka, odnosno neravnoteža klasa je, kao što je već prethodno rečeno, uobičajeni problem kada govorimo o medicinskim slikama. Vrlo često se dešava da postoji mali broj dostupnih podataka za manjinsku klasu koja je ujedno i klasa od interesa, jer predstavlja klinički važno stanje. Kada većinska klasa dominira skupom podataka, konvolucijske neuronske mreže imaju tendenciju da postanu pristrasne, što rezultuje slabom osetljivošću za manjinsku klasu. Radi rešavanja ovog problema razvijene su razne tehnike, koje se mogu grupisati u tehnike na nivou modela, na nivou podataka i na nivou algoritma, a najrelevantnije biće detaljnije opisane u nastavku ovog poglavlja.

Pretreniranje (engl. overfitting) i problem nedovoljne količine podataka

Neravnoteža klasa često dovodi do pretreniranja, posebno na ograničenim podacima koji pripadaju manjinskoj klasi. Problem pretreniranja ili *overfitting* nastaje kada model mašinskog učenja upozna sve detalje skupa trening podataka, što znači da ima odlične rezultate nad skupom trening podataka, ali generalno loš rezultat pri obradi podataka koje nije prethodno video. Ovo je jako čest problem u domenu mašinskog učenja i postoji dosta metoda koje pokušavaju da utiču na neutralizaciju ili sprečavanje nastanka ovog problema. Model mašinskog učenja u nekom trenutku počinje da prepoznaše obrasce koji su specifični za skup trening podataka, ali navode na pogrešne zaključke ili su potpuno irelevantni kada je reč o novim podacima. Takođe, kao što se može videti na SlikaXZ, postoji i *underfitting*, pojava suprotna *overfitting-u* koja pokazuje da mreža nije dovoljno složena za rešavanje datog problema.



SlikaXZ. Prikaz underfitting, overfitting i dobro balansiranog skupa podataka

Najbolji način koji pomaže u prevazilaženju problema pretreniranja je veći skup trening podataka, ali kako to nije uvek moguće obezbediti razvijeno je nekoliko metoda:

- **Regularizacija:** Jedno od rešenja je moduliranje količine informacija koje model sme da čuva ili dodavanje ograničenja za čuvanje informacija. Ukoliko ima manje obrazaca za pamćenje, mreža će se prilagoditi da pamti što relevantnije podatke, koji pomažu pri generalizaciji. U računanje same funkcije gubitka dodaje se regularizacioni član, a najčešće se koriste L1 i L2 tipovi regularizacije;
- **Dropout slojevi:** Tehnika u kojoj se određeni broj nasumično odabralih neurona zanemaruje tokom treninga. Ovo znači da oni neće imati doprinos aktivaciji neurona u nižim slojevima i da njihova težina neće biti ažurirana putem backpropagacije;
- **Batch Normalization slojevi:** Tehnika koja se sastoji od dodavanja operacije u model neposredno pre funkcije aktivacije svakog sloja i vrši standardizaciju ulaza u sloj za svaku mini-seriju podataka;
- **Rano zaustavljanje** (*engl. Early Stopping*): Ova tehnika podrazumeva da se treniranje mreže zaustavlja kada metrika prestane da napreduje;
- **Augmentacija podataka:** Još jedan od načina rešavanja problema *overfitting-a*, posebno kod modela koji obrađuju slike, a biće detaljnije objašnjeno u nastavku.

Prethodno navedene tehnike pomažu u ublažavanju pretreniranja, ali je pored njih takođe neophodno primeniti i specifične metode za neravnotežu klasa (opisane u nastavku) kako bi se osiguralo da model pravedno tretira sve klase.

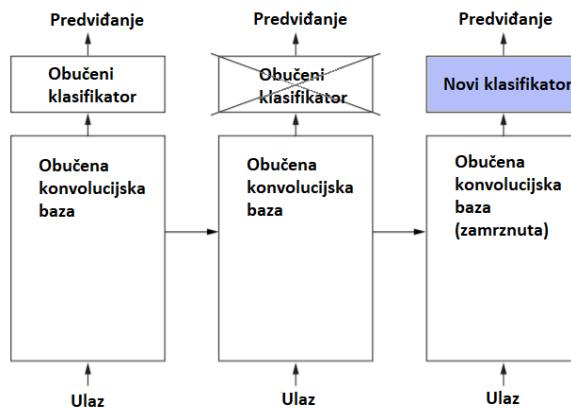
Transfer učenja

U praksi se veoma retko dešava da se trenira cela konvolucijska neuronska mreža od nule, jer duboke neuronske mreže zahtevaju veliku količinu podataka, koju često nije moguće obezbediti, naročito kada govorimo o medicinskim slikama. Zbog toga se u kontekst neuronskih mreža, uvodi pojam transfer učenja (*eng. transfer learning*) koji upućuje na razmenu znanja između različitih modela neuronskih mreža. Podrazumeva se da je prethodno obučena mreža bila trenirana na jako velikom skupu podataka, obično na zadatku klasifikacije slika. Ako je ovaj izvorni skup podataka dovoljno veliki i dovoljno opšti, tada naučena prostorna hijerarhija obrazaca (ovaj je pojam već prethodno objašnjen) može efikasno delovati kao generički model vizuelnog sveta, pa se stoga njegove karakteristike mogu pokazati korisnim za mnoge različite probleme računarskog vida, iako novi problemi mogu da uključuju potpuno različite klase od onih iz prvobitnog zadatka. Ovo smanjuje pretreniranje i poboljšava prepoznavanje manjinske klase, jer mreža počinje sa opštim vizuelnim karakteristikama umesto da uči u potpunosti iz ograničenih medicinskih podataka.

Transfer učenja može se klasifikovati na tri različita:

- **Induktivni** - zadaci izvorni i ciljni su različiti, bez obzira da li su domeni isti ili nisu;
- **Transduktivni** - zadaci izvorni i ciljni su isti, dok su domeni različiti;
- **Nenadgledani** - zadaci izvorni i ciljni su različiti, dok su domeni isti.

Kao što je već rečeno, konvolucijske neuronske mreže koje se koriste za klasifikaciju slika sastoje se iz dva dela: serije slojeva konvolucije i sažimanja, a završavaju se gusto povezanim klasifikatorom. Prvi deo naziva se konvolucionima osnova modela. Izdvajanje karakteristika se sastoji iz uzimanja konvolucijske osnove prethodno obučene mreže, provođenju novih podataka kroz nju i treniranju novog klasifikatora na vrhu kao što je prikazano na Slika XX. [22]



Slika XX: Zamena klasifikatora uz zadržavanje iste konvolucijske baze

Postoje dve najčešće korišćene strategije za iskorišćavanje prethodno obučene mreže: [19]

- Prva strategija se naziva **izdvajanje karakteristika putem transfera učenja** i odnosi se na pristup u kome prethodno obučeni model zadržava i svoju početnu arhitekturu i sve naučene težine. Dakle, prethodno obučeni model se koristi samo za izdvajanje karakteristika koje se zatim ubacuju u novu mrežu koja obavlja zadatak klasifikacije;
- Druga strategija se odnosi na sofisticiraniju proceduru, u kojoj se primenjuju **specifične modifikacije** (podešavanja arhitekture i/ili podešavanja parametara) **na prethodno obučeni model**, kako bi se postigli optimalni rezultati. Na ovaj način se zadržava samo specifično znanje iz prethodnog zadatka, dok se u mrežu ubacuju novi parametri za obuku. Novi parametri zahtevaju treniranje na relativno velikoj količini podataka da bi bili korisni.

Važno je napomenuti da slojevi koji dolaze ranije u modelu izdvajaju lokalnu, vrlo generičku mapu karakteristika (kao što su vizuelne ivice, boje i teksture), dok viši slojevi izvlače apstraktnije koncepte (poput „mačje uho“ ili „pseće oko“). Dakle, ako se novi skup podataka

mnogo razlikuje od skupa na kojem je originalni model treniran, potrebno je koristiti samo prvih nekoliko slojeva modela za izdvajanje karakteristika, umesto cele konvolucijske baze.

Transfer učenja se može postići upotrebom jednog od tri moguća scenarija: [37]

- a) **Plitko podešavanje** (*engl. Shallow Tuning*) prilagođava samo poslednji sloj klasifikacije da se nosi sa novim zadatkom, i zamrzava parametre preostalih slojeva bez obuke;
- b) **Duboko podešavanje** (*engl. Deep Tuning*) ima za cilj da ponovo obuči sve parametre unapred obučene mreže;
- c) **Fino podešavanje** (*engl. Fine Tuning*) ima za cilj da postepeno trenira više slojeva, podešavanjem parametara učenja, dok se ne postigne značajno povećanje performansi. Prenos znanja putem mehanizma finog podešavanja pokazao je izvanredne performanse u klasifikaciji rendgenskih snimaka grudnog koša.

Fino podešavanje

Fino podešavanje je široko korišćena tehnika za ponovnu upotrebu modela. Kod ovog scenarija se vrši odmrzavanje nekoliko gornjih slojeva zamrznute konvolucijske baze prethodno obučenog modela koji se koriste za izdvajanje karakteristika, a zatim se vrši zajedničko treniranje novog modela i gornjih slojeva prethodno obučenog modela. Ova tehnika se naziva fino podešavanje, jer neznatno prilagođava apstraktnije delove modela koji se ponovo koristi, kako bi ih učinio relevantnim za trenutni problem. [45]

Najpre je neophodno zamrznuti konvolucijsku bazu prethodno obučenog modela kako bi slučajno inicijalizovan klasifikator prešao na vrh i bio treniran. Nakon toga je moguće fino podešavati slojeve koji se nalaze na vrhu konvolucijske baze prethodno obučenog modela. U slučaju da pomenuti klasifikator nije istreniran, greške koje bi se propagirale kroz mrežu bile bi prevelike, a reprezentacija rezultata na osnovu prethodno naučenih slojeva uništena. Zbog toga je za pravilno fino podešavanje prethodno obučene mreže, neophodno preuzeti sledeće korake:

1. Dodavanje slučajno inicijalizovanog klasifikatora na vrh konvolucijske baze prethodno obučenog modela;
2. Zamrzavanje konvolucijske baze prethodno obučenog modela;
3. Treniranje dela koji se dodaje;
4. Odmrzavanje određenih slojeva konvolucijske baze prethodno obučenog modela;
5. Treniranje odmrznutih slojeva i dodatog dela zajedno.

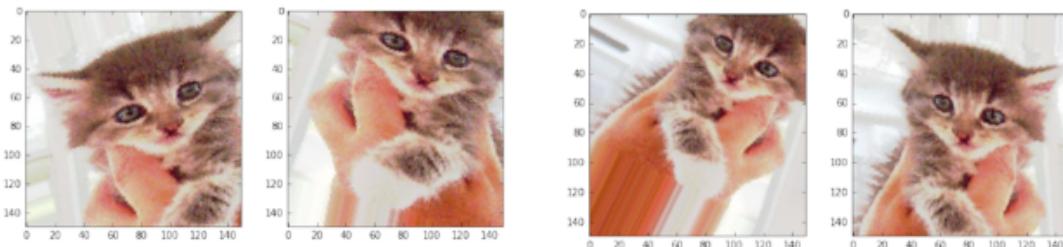
Osnovna konvolucijska neuronska mreža sadrži preko petnaest miliona parametara (neurona), a što više parametara je obuhvaćeno tokom treniranja, utoliko je veća šansa da dođe do

pretreniranja, pogotovu na malom skupu podataka, te se zbog toga za fino podešavanje biraju samo dva ili tri sloja koja se nalaze na vrhu.

Tehnike na nivou podataka

Augmentacija podataka

Kao što je prethodno pomenuto, u slučaju da postoji nebalansirani skup podataka overfitting će se sigurno javiti, te augmentacija podataka (*engl. data augmentation*) predstavlja jedan od načina prevazilaženja ovog problema. Ovaj metod ima poseban značaj kod obrade slika uz pomoć nekog od modela dubokog učenja. [22]



Slika YZ. Primer augmentacije na slici mačke

Augmentacija podataka koristi postojeće podatke iz skupa trening podataka i utiče do određene mere na njih, primenom transformacija, kako bi proizvela nove podatke za skup trening podataka. Cilj augmentacije je da, uoči treniranja, model uvek dobije različitu sliku na kojoj se trenira, a jednostavne transformacije (rotacija, translacija, podešavanje osvetljenja i kontrasta, skaliranje i isecanje, dodavanje šuma i sl.) to omogućavaju. Augmentacija je posebno korisna za manjinsku klasu jer efikasno proširuje njenu reprezentaciju u skupu trening podataka.

Kada treniramo mrežu uz pomoć posebne konfiguracije koja omogućava augmentaciju podataka, ona nikada neće videti isti ulazni podatak i biće naučena da bolje vrši generalizaciju. Pošto se u slučaju augmentacije podataka koristi ista slika kao osnova, to će uticati do odredene mere na model, ali on neće biti u stanju da donosi nove zaključke i zbog toga sama metoda nije dovoljna u svim slučajevima za prevazilaženje problema overfitting-a skupa podataka.

Naduzorkovanje podataka

Tehnikama na nivou podataka za rešavanje problema nedovoljne količine podataka pripadaju **naduzorkovanje** (*engl. over-sampling*) i **poduzorkovanje** (*engl. under-sampling*). Ove tehnike vrše modifikaciju skupa trening podataka kako bi smanjile nivo neravnoteže ili kako bi smanjile

šum (pogrešno označene uzorke ili anomalije). U svojim najosnovnijim oblicima naduzorkovanje duplira slučajno odabранe uzorke manjinske klase, dok poduzorkovanje odbacuje slučajno odabranе uzorke većinske klase. [30]

Dakle imamo da poduzorkovanje namerno odbacuje podatke, smanjujući ukupnu količinu informacija iz kojih model uči. To je i glavni razlog zbog koga se poduzorkovanje uglavnom ne koristi u medicinskom domenu, jer dovodi do gubitka teško dostupnih i klinički važnih podataka.

Suprotno tome, naduzorkovanje povećava vreme obuke modela zbog povećanog skupa za treniranje, ali je utvrđeno da može izazvati i pretreniranje, pa je stoga razvijen niz inteligentnih metoda naduzorkovanja u pokušaju da se uravnoteže ovi kompromisi.

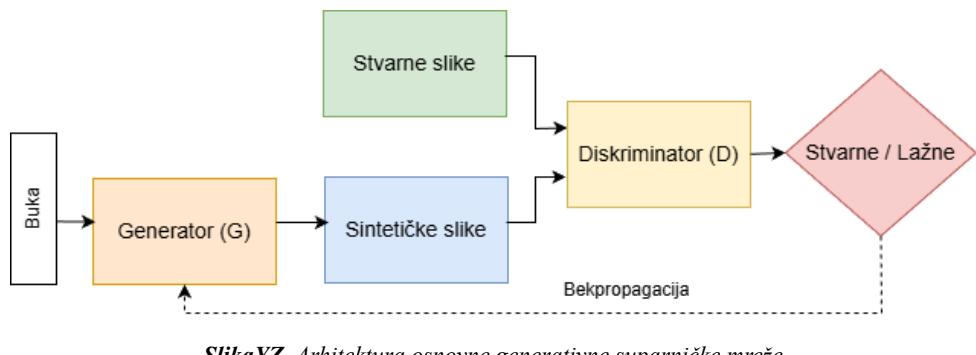
Najpoznatija inteligentna metoda naduzorkovanja je metoda sintetičkog manjinskog naduzorkovanja ili skraćeno **SMOTE metoda** (*engl. Synthetic Minority Over-sampling Technique*) koja proizvodi veštačke uzorke manjinske klase interpolacijom između postojećih manjinskih instanci. Međutim, SMOTE metoda se ne preporučuje za medicinske slike zbog stvaranja nerealnih vizuelnih artefakta. [57]

Sa druge strane, **sintetičke slike generisane pomoću generativne suparničke mreže (skr. GAN)** mogu se koristiti kao inteligentna metoda naduzorkovanja za rešavanje problema disbalansa klasa kod medicinskih slika, jer mogu generisati realistične patološke slike. [55] Način njihovog rada biće detaljnije opisan u nastavku.

Generisanje sintetičkih podataka korišćenjem GAN-a

Jedan od najnovijih i najefikasnijih pristupa za ublažavanje problema neravnoteže klasa u skupovima podataka slika je upotreba generativne suparničke mreže (*skr. GAN*). Poslednjih godina, ove mreže su privukle veliku pažnju istraživača raznih oblasti primene zbog svoje sposobnosti da kreiraju nove, realistične podatke učeći iz postojećih uzoraka.

Generativne suparničke mreže predložili su Ian Goodfellow i saradnici 2014. godine [51] kao novi duboki generativni model za generisanje slika. Originalna arhitektura GAN-a podrazumeva postojanje dve mreže, generatora i diskriminatora, koje se istovremeno treniraju, kao što je prikazano na SlikaYZ. Zadatak generatorske mreže je da nauči da proizvede sintetički podatak koji izgleda kao da je stvarni, dok diskriminator pokušava da prepozna koji podaci su sintetički (lažni). Kroz treniranje generatorska mreža postepeno postaje sposobna da generiše nove, uverljive podatke, dok se diskriminator sve više muči da razlikuje stvarne od lažnih, što pokazuje da je GAN dostigao dobro obučeno stanje. [55]



SlikaYZ. Arhitektura osnovne generativne suparničke mreže

U literaturi se pominje širok spektar GAN mreža, svaka sa svojim prednostima u ublažavanju problema klasne neravnoteže u zadacima računarskogvida, a neke od njih, koje se uglavnom koriste za generisanje sintetičkih podataka kod problema klasifikacije su:

- **CGAN** (*engl. Conditional Generative Adversarial Networks*) - kod uslovne generativne suparničke mreže dodaje se uslovni parametar koji kontroliše proces generisanja, što omogućava proizvodnju određene vrste izlaza (kontrola vrste podataka koji se generišu);
- **ACGAN** (*engl. Auxiliary Classifier Generative Adversarial Networks*) - Pomoćna klasifikatorska generativna suparnička mreža predstavlja proširenje CGAN arhitekture, kod koje diskriminator dobija samo sliku, za razliku od CGAN-a koji dobija i sliku i oznaku klase kao ulaz. Modifikovan je da razlikuje stvarne i lažne podatke, kao i da rekonstruiše označe klase;
- **DCGAN** (*engl. Deep Convolutional Generative Adversarial Networks*) - Duboka konvolucijska generativna suparnička mreža koristi konvolucione i transponovano-konvolucione slojeve u diskriminatoru i generatoru, respektivno;
- **WGAN** (*engl. Wasserstein Generative Adversarial Network*) - Vaserštajnova generativna suparnička mreža koristi, takozvano, Vaserštajnovo rastojanje (WD), koje pruža smislenu i glađu meru rastojanja između distribucija. Od klasičnog GAN-a razlikuje se po funkciji greške i gradijentnoj kazni;
- **CycleGAN** - Kod ove arhitekture se koriste dva generatora i dva diskriminadora za obavljanje zadataka prevođenja neuparenih slika na slike. CycleGAN usvaja koncept ciklične konzistentnosti iz mašinskog prevođenja, gde rečenica prevedena sa engleskog na španski i prevedena nazad sa španskog na engleski treba da budu identične.

Generative suparničke mreže predstavljaju moćno proširenje tehnika naduzorkovanja, posebno efikasnih za klasifikaciju slika u uslovima jake neravnoteže klase. Ova tehnika, ne samo da povećava reprezentaciju manjinske klase, već može pomoći i u sprečavanju pretreniranja. Međutim, sintetičke slike se moraju koristiti sa oprezom, jer ako kvalitet sintetičkih slika nije

visok, to bi dovelo do dodatnog šuma u originalnim skupovima podataka. Sintetičke slike generisane GAN-om ne mogu se koristiti kao potpuna zamena za stvarne skupove podataka, ali mešavina stvarnih i GAN-om generisanih slika ima ogroman potencijal da poveća performanse modela dubokog učenja.

Tehnike na nivou algoritma

Ponderisana kategorička unakrsna entropija

U zadacima klasifikacije medicinskih slika, kada se koriste konvolucijske neuronske mreže, kao standardna funkcija greške uzima se kategorička unakrsna entropija o kojoj je ranije već bilo reči. Međutim, ona nije optimalna kada se radi sa nebalansiranim skupom podataka, jer prepostavlja da sve klase podjednako doprinose ukupnoj grešci. U takvim slučajevima, model ima tendenciju da bude pristrasan ka većinskim klasama, jer njihove greške dominiraju u proračunu gubitaka.

Da bi se prevazišao ovaj problem, uvedena je ponderisana kategorička unakrsna entropija (*engl. Weighted Categorical Cross-Entropy*), koja prilikom izračunavanja greške uzima u obzir i težinu koja je dodeljena klasi kojoj pripada uzorak za obuku. Težine koje se dodeljuju klasama su obrnuto proporcionalne učestalosti klase, što osigurava da model posvećuje veću pažnju na nedovoljno zastupljene kategorije tokom obuke i sprečava da gubitkom dominira većinska klasa. Težine za svaku klasu i ponderisana kategorička unakrsna entropija (*skr. WCE*) računaju se korišćenjem sledećih jednačina: [41]

$$w(c) = C_c * \frac{\sum_{c=0}^N n_c}{N * n_c} \quad \text{WCE} = -\sum_i^C w(i) * t_i * \log(p_i)$$

gde je za prvu formulu C_c konstanta za klasu c, N je broj klasa, a n_c broj uzoraka u klasi c, dok je za drugu formulu C broj klasa, w(i) težinski faktor, t_i istinita klasa, a p_i predstavlja predviđenu verovatnoću za istinitu klasu.

Ova funkcija greške povećava kaznu za pogrešnu klasifikaciju manjinskih klasa, pomažući modelu da nauči uravnotežene granice odlučivanja.

Fokalni gubitak

Fokalni gubitak (*engl. Focal loss*) je još jedna funkcija greške koja je sofisticiranija u odnosu na ponderisanu kategoričku unakrsnu entropiju, jer pored uzimanja u obzir težina dodeljenih

klasama, fokusira učenje na uzorke koje je teško klasifikovati. Lin i njegovi saradnici [56] predložili su fokalni gubitak kao alternativnu funkciju greške koja ima za cilj da reši problem neravnoteže klasa, a računa se korišćenjem sledeće jednačine: [40]

$$\text{Fokalni gubitak} = -\alpha_c (1 - p_c)^\gamma \log(p_c)$$

u kojoj p_c predstavlja predviđenu verovatnoću za istinitu klasu, α_c predstavlja faktor težine, a γ je parametar fokusiranja (smanjuje gubitak kod jednostavnih primera).

Kako je klasična funkcija fokalnog gubitka prvenstveno dizajnirana za binarne probleme, za slučaj višestruke klasifikacije može se koristiti njena unapređena verzija: [40]

$$\text{Fokalni gubitak} = -\sum_i^C \alpha_c (1 - p_c)^\gamma t_i \log(p_c)$$

gde C predstavlja broj klasa, a t_i istinitu klasu.

Fokalni gubitak stavlja naglasak na uzorke (slike) koje pripadaju manjinskoj klasi i koje je teško kategorizovati, smanjujući pristrasnost ka većinskim klasama. Ovo pomaže u poboljšanju učinka modela u identifikaciji nedovoljno zastupljene klase i na taj način pomaže rešavanju problema nedovoljne količine podataka.

Hibridne tehnike

Hibridne tehnike za prevazilaženje problema nedovoljne količine podataka kombinuju tehnike na nivou podataka i tehnike na nivou algoritma na različite načine, a sa ciljem ublažavanja problema nedovoljne količine podataka i disbalansa klasa. [30]

Ovakav pristup uključuje naduzorkovanje podataka kako bi se smanjila neravnoteža klasa, odnosno generisali novi uzorci za manjinsku klasu, a zatim primenu neke od algoritamskih tehnika, kako bi se dodatno smanjila pristrasnost prema većinskoj klasi. Na primer, kada se kombinuju sintetičke slike, generisane korišćenjem GAN-a, sa ponderisanom kategoričkom unakrsnom entropijom, modeli postižu bolju stabilnost, manju pristrasnost i poboljšanu generalizaciju.

Hibridne tehnike istovremeno povećavaju raznovrsnost i reprezentativnost podataka, dok se model usmerava da bolje uči karakteristike manjinskih klasa. Međutim, uspeh hibridnih metoda zavisi od kvaliteta generisanih podataka i pravilnog odabira težina za funkciju greške.

Korišćene tehnologije

U ovom poglavlju opisani su programski jezici, platforme, biblioteke i okruženja koji su korišćeni za praktičnu realizaciju rada.

WSL (Windows podsistem za Linux)

WSL je kompatibilni sloj koji omogućava korišćenje Linux operativnog sistema na Windows operativnom sistemu bez potrebe za virtuelnom mašinom. Pruža izvorni pristup Linux alatima i bibliotekama, što je posebno korisno u slučaju mašinskog učenja koje zahteva Linux-specifičnu podršku za grafički procesor. Korišćenje WSL-a omogućava bolju kompatibilnost sa programskim okruženjima za duboko učenje i pojednostavljuje postavku za treniranje modela korišćenjem grafičkog procesora.

Visual Studio Code

Visual Studio Code je alat za uređivanje koda koji je razvio Microsoft za operativne sisteme Windows, Linux i macOS. Podržava brojne programske jezike i nudi bogatu listu ekstenzija za otklanjanje grešaka, kontrolu verzija i upravljanje okruženjem. Obuhvata integrисани terminal i ima podršku za udaljeni razvoj, što uključuje WSL.

Python

Python je programski jezik interpretorskog tipa, objektno-orientisani jezik visokog nivoa, interaktivni, poseduje dinamičku semantiku i opšte je namene. Nastao je 1991. godine, stvorio ga je Gvido van Rossum (*engl. Guido van Rossum*), a ime je dobio po kulturnoj britanskoj komediji „Monty Python“. Akcenat kod ovog jezika je na njegovoj jednostavnoj sintaksi, preglednosti koda i velikoj fleksibilnosti.

Anaconda (Miniconda)

Anaconda je besplatna distributivna platforma otvorenog koda za jezike Python i R. Ova platforma koristi koncept kreiranja promenljivih okruženja, tako da može da izoluje različite biblioteke i njihove verzije. Obuhvata veliki broj paketa kao što su conda, numpy, scipy, ipython, notebook i mnoge druge, a može se koristiti na Windows, Linux i macOS operativnim sistemima. Miniconda je besplatna, minijaturna instalacija Anaconda platforme koja uključuje znatno manju količinu paketa, što je čini bržom i jednostavnijom za kontrolu.

TensorFlow

TensorFlow je softverska biblioteka otvorenog koda za numerička izračunavanja, a posebno je pogodna i prilagođena za jako velike zadatke iz oblasti mašinskog učenja. Princip rada ove biblioteke podrazumeva da se u programskom jeziku Python definiše graf operacija koje je

potrebno obaviti, a zatim TensorFlow uzima taj graf i efikasno ga izvršava koristeći optimizovani C++ kod. [23] Još važnije je to da je moguće rasparčati graf operacija na nekoliko delova i izvšavati ih paralelno na većem broju CPU-a ili GPU-a.

Neke od najzanimljivijih odlika TensorFlow-a date su u nastavku [23]:

- Razvio ga je Google Brain tim stručnjaka i on pokreće Google-ove servise kao što su Google Cloud Speech, Google Photos i Google Search;
- Može se pokrenuti i na mobilnim uređajima i to i na IOS-u i na Android-u;
- Njegov osnovni Python API nudi dosta fleksibilnosti u kreiranju svih vrsta izračunavanja, uključujući bilo koju arhitekturu neuronskih mreža;
- Uključuje veoma efikasnu C++ implementaciju za veliki broj operacija mašinskog učenja, kao i C++ API za definisanje korisničkih operacija visokih performansi;
- Pruža nekoliko naprednih čvorova za optimizaciju radi traženja parametara koji minimizuju troškove operacija. Jako su jednostavnii za upotrebu, jer TensorFlow automatski računa gradijent funkcija koje korisnik definiše (automatsko diferenciranje).

Keras

Keras je biblioteka otvorenog koda za neuronske mreže napisana u programskom jeziku Python. Dizajnirana je tako da omogući brzo eksperimentisanje sa dubokim neuronskim mrežama, a fokus je na tome da bude jednostavna za korišćenje, modularna i proširljiva.

Neke od najzanimljivijih karakteristika ove biblioteke date su u nastavku [54]:

- Keras je najviše korišćeno programsko okruženje za duboko učenje od top-5 pobedničkih timova na Kaggle-u. Razlog je taj što Keras olakšava izvođenje novih eksperimenata i omogućava brže isprobavanje više ideja od konkurencije;
- Kako je Keras izgrađena nad Tensorflow bibliotekom, predstavlja jako snažno okruženje koje može da se skalira na ogromne klasterne GPU-a ili na čitav TPU pod⁹ i to vrlo jednostavno;
- Keras modeli se mogu eksportovati na JavaScript i izvršavati direktno u pretraživaču ili na TF Lite¹⁰ radi izvršavanja na IOS-u, Androidu ili embedded uređajima;
- Keras koriste CERN i NASA, kao i mnoge druge naučne organizacije u svetu, jer ima veliku fleksibilnost čak i na najnižim nivoima za implementaciju proizvoljnih istraživačkih ideja, dok nudi opcione pogodnosti na visokom nivou radi ubrzavanja ciklusa eksperimentisanja.

⁹ **TPU pod** je konfiguracija u Google-ovom centru za podatke koji ima niz TPU (*engl. Tensor Processing Unit*) uređaja međusobno povezanih veoma brzom mrežom. Kada je koristimo za izvršavanje sva izračunavanja mašinskog učenja koja su nam potrebna se distribuiraju na sve TPU uređaje.

¹⁰ **TF Lite** (*engl. TensorFlow Lite*) je skup alata koji programerima omogućava izvršavaju TensorFlow modela na mobilnim, embedded i IoT (*engl. Internet of Things*) uređajima.

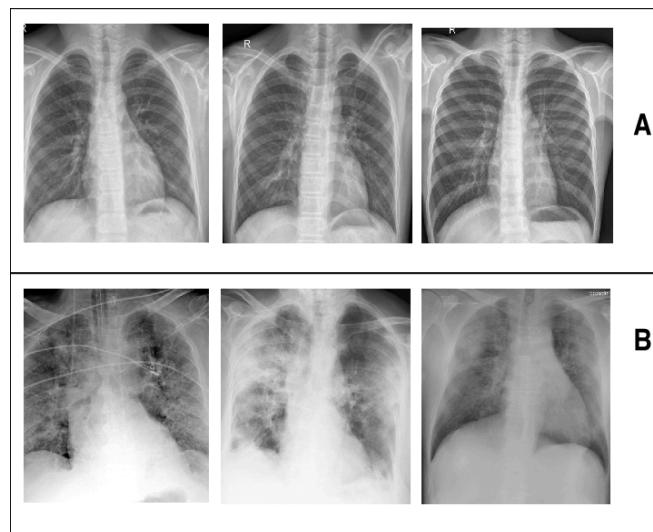
Implementacija

Od pojave novog korona virusa mnogo naučnika je radilo i još uvek radi na ispitivanju efikasnosti tehnika mašinskog učenja, konkretnije konvolucijskih neuronskih mrež za detekciju ovog oboljenja korišćenjem CXR slika pluća. Radovi realizovani u prvim godinama od pojave virusa, uključujući i one navedene u odeljku “*Povezani radovi*”, su prilikom svojih ispitivanja korisili ekstremno malu količinu slika za slučaj pluća sa potvrđenim COVID-19 oboljenjem.

Ideja ovog rada je da se utvrdi koliko su tehnike za prevazilaženje nedovoljne količine podataka zaista efikasne kada se radi sa ekstremno malim skupovima podataka. Stoga će konvolucijske neuronske mreže biti trenirane i validirane na jednom, jako malom skupu podataka, uz upotrebu nekoliko tehnika za prevazilaženje problema nedovoljne količine podataka, a zatim testirane na drugom, većem skupu. U nastavku, najpre će biti reči o COVID-19 karakterističnim abnormalnostima, vidljivim na radiografskim snimcima, zatim će biti opisani korišćeni skupovi podataka, detalji o delovima implementacije koji su najznačajniji, a zatim i rezultati istraživanja.

Specifične karakteristike oboljenja

Da bi se uopšte moglo razmišljati o upotrebi veštačke inteligencije za detekciju ili klasifikaciju neke bolesti korišćenjem medicinskih slika, neophodno je dokazati da postoje neke abnormalnosti, koje su vidljive na radiografskim snimcima, a koje su jedinstvene za to oboljenje.



*Slika XX: Poređenje slika čistih i pluća sa COVID-19 infekcijom.
Skup A su čista, a skup B COVID-19 pluća.*

Medicinska istraživanja su pokazala da se kod pacijenata, koji su inficirani COVID-19 virusom, mogu pomoći radiografskim snimakama uočiti karakteristične abnormalnosti, čak i u slučaju odsustva uobičajenih simptoma i to vrlo brzo nakon zaražavanja [38, 39]. Na primer, Huang i drugi [12] identifikovali su da većina COVID-19 pozitivnih slučajeva iz njihovog istraživanja imaju bilateralne radiografske abnormalnosti u CXR slikama, dok su Guan i ostali [13] takođe uočili abnormalnosti u vidu "mlečnog stakla" (*engl. ground-glass opacity - GGO*), bilateralnih i intersticijskih (međuprostorna tečnost) abnormalnosti na CXR i CT slikama.

Naime, kada bolest napreduje u telu pacijenta pozitivnog na COVID-19 virus, dolazi do pojave tečnosti u malim vazdušnim vrećicama u plućima zvanima alveole [15]. Prisustvo ove tečnosti izaziva zapaljenje pluća, a napredak upale može se pratiti preko CXR i CT slika. Upala pluća se pojavljuje u vidu "mlečnih stakala", a njihovim spajanjem nastaju i "mlečne konsolidacije".

Izbor vrste radiografskih snimaka

Kao što je već prethodno rečeno, postoje dve mogućnosti kada govorimo o radiografskim snimcima, a to su CXR i CT slike pluća. CXR slike se koriste radi ispitivanja gustih tkiva, dok u slučaju CT slika, imamo detaljniji prikaz grudnog koša, te se mogu ispitivati i kosti, meka tkiva i krvni sudovi. [16] Zbog ovih razlika se kroz dosta radova provlači polemika oko izbora vrste radiografskih snimaka za detekciju COVID-19 virusa, a karakteristike koje su bile presudne za najčešći izbor CXR snimaka su [20]:

- **Brza trijaža¹¹** - CXR snimci omogućavaju brzu trijažu pacijenata za koje se sumnja da su zaraženi, jer je za njihovo snimanje potrebno svega par minuta;
- **Dostupnost i pristupačnost** - CXR snimanje je dostupno u većini kliničkih centara i centara za snimanje, a pored toga smatra se standardnom opremom zdravstvenih sistema;
- **Prenosivost** - Postojanje prenosivih CXR sistema znači da se snimanje može izvoditi i na terenu, što nije slučaj sa CT skenerom koji predstavlja fiksni sistem;
- **Manja doza zračenja** - Radiografija pluća podrazumeva dozu zračenja od $0,05\text{mSv}^{12}$, dok kompjuterizovana tomografija abdomena dostiže dozu zračenja od 10mSv .

U praktičnom delu ovog rada ispitivanja će se vršiti na CXR slikama pluća i iz prethodno navedenih razloga, a takođe i zbog toga što je to slučaj najvećeg broja radova koji su se bavili proučavanjem teme detekcije i klasifikacije infekcije COVID-19.

¹¹ **Trijaža** je jedan od medicinskih sistema razvrstavanja pacijenata koji se primenjuje u uslovima masovnih događaja (velike nesreće, pandemije i sl), kako bi se omogućilo brzo prepoznavanje osoba koje će imati najveću korist ranog zbrinjavanja i brzog transporta u odgovarajuću zdravstvenu ustanovu.

¹² **Sivert** je SI izvedena jedinica ekvivalentne doze radijacije i kao takva je zavisna od bioloških efekata radijacije, nasuprot fizičkim aspektima, koje karakteriše apsorbovana doza koja se meri u grejima.

Korišćeni skupovi podataka

Detekcija i klasifikacija bolesti na osnovu CXR snimaka pluća korišćenjem mašinskog učenja je dosta popularna tema, te zahvaljujući tome na internetu postoji veliki broj dostupnih skupova podataka. Za istraživanja ovakvog tipa bilo je potrebno izdvojiti pouzdane izvore, pa su nakon šire analize izdvojeni [17], [50] i [58, 59, 60, 61, 62] skupovi.

Prvi skup podataka [17] delo je Dr. Joseph Cohen-a i njegovih saradnika, istraživača sa Univerziteta Stanford i sadrži CXR i CT slike pluća za koje je potvrđena infekcija virusom COVID-19 ili neka druga bakterijska/virusna infekcija. Ovo je inicijativa pokrenuta sa ciljem da se prikupe CXR i CT slike pluća sa potvrđenom COVID-19 infekcijom i objave tako da budu javno dostupne za naučna ispitivanja zasnovana na veštačkoj inteligenciji. Podatke su prikupljali sa online publikacija, veb sajtova kao i PDF-ova pomoću različitih alata sa ciljem da se tokom tog procesa održi kvalitet slika. Svi radovi navedeni u odeljku “*Povezani radovi*” koriste ovaj skup podataka kao primarni izvor slika sa potvrđenom COVID-19 infekcijom. Za ovo istraživanje je iz skupa izdvojeno 196 CXR slika sa potvrđenom COVID-19 infekcijom, a koje su slikane u uspravnom položaju pacijenta (skr: PA¹³).

Drugi skup podataka [50] je skup medicinskih slika prikupljenih u periodu od 1992 do 2015 godine sa Kliničkog centra Nacionalnog instituta za zdravlje Amerike. U pitanju je veliki skup sa ukupno 112.120 CXR slika pluća i odgovarajućim pridruženim metapodacima koji su korišćenjem NLP¹⁴ tehnike izdvojeni iz radioloških izveštaja. Za ovo istraživanje je iz skupa izdvojeno 630 slika čistih pluća i 630 slika pluća sa detektovanom pneumonijom, a koje su slikane u uspravnom položaju pacijenta.

Treći skup podataka [58, 59, 60, 61, 62] kreirala je grupa istraživača sa Univerziteta u Kataru i predstavlja mešavinu podataka prikupljenih sa više javno dostupnih repozitorijuma i kliničkih centara. Jedan od izvora je skup [17], drugi čine podaci prikupljeni sa dobro poznatog Kaggle takmičenja za prepoznavanje pneumonije pluća (*engl. RSNA Pneumonia Detection Challenge*), a još jedan važan izvor su i snimci pluća pacijenata San Juan bolnice u Španiji prikupljeni u periodu od 2009 do 2017 godine. Ovaj skup sadrži 33.920 rendgenskih snimaka pluća raspodeljenih u tri klase COVID-19, pneumonia i čista pluća. Za ovo istraživanje je iz skupa izdvojeno 1712 slika čistih pluća, 1802 slika pluća sa pneumonijom (bakterijska i virusna) i 1903 slika sa potvrđenom COVID-19 infekcijom, a koje su slikane u uspravnom položaju pacijenta.

¹³ PA (posterior-anterior) - rendgensko slikanje pluća kod koga je pacijent u stojećem položaju sa grudima prislonjenim na film, a emiterom zraka iza leđa.

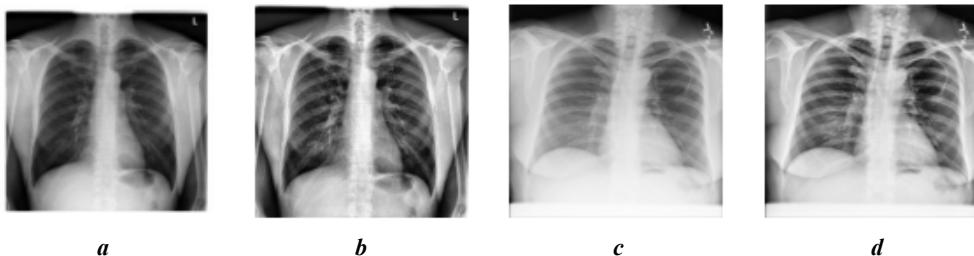
¹⁴ NLP (*engl. Natural Language Processing*) je polje veštačke inteligencije koje se bavi obradom prirodnih jezika.

Učitavanje i preprocesiranje

Da bi se podaci pripremili za pohranjivanje neuronske mreže obavljeno je preprocesiranje kroz nekoliko koraka. Korišćene su uslužne metode Keras biblioteke i njen modul slojeva (*engl. tensorflow.keras.layers*), metode Tensorflow biblioteke, kao i metode OpenCV biblioteke.

Prilikom preuzimanja podataka iz izvornih repozitorijuma i njihovog učitavanja u direktorijum za treniranje i validaciju, primenjene su tehnike poboljšanja kontrasta i izdvajanja plućnog polja. Na ovaj način uklonjena je neželjena pozadinska buka i sprečeno da model donosi zaključke na osnovu irelevantnih delova slike.

Za poboljšanje kontrasta slika korišćena je **CLAHE (Adaptivno izjednačavanje histograma sa ograničenim kontrastom)** metoda OpenCV biblioteke koja je varijanta AHE (Adaptivno izjednačavanje histograma) metode. CLAHE metoda vodi računa o prekomernom pojačavanju kontrasta, jer radi na malim regionima na slici, koji se nazivaju pločice, umesto na celoj slici. U odnosu na AHE računarski je dosta skuplja, ali se pokazala kao dobra opcija u slučaju rada sa CXR slikama pluća, jer poboljšava performanse rada konvolucijske neuronske mreže [52].



Slika XX: Primeri CXR slika pluća pre (a,c) i nakon (b,d) primene CLAHE algoritma

```
def apply_clahe(image):
    clahe = cv2.createCLAHE(clipLimit=2.0, tileSize=(8, 8))
    clahe_image = clahe.apply(image)
    return clahe_image
```

Slika XX: Funkcija korišćena za primenu CLAHE algoritma

Izdvajanje plućnog polja obavljeno je pomoću **YOLO (engl. You Only Look Once)** algoritma za detekciju regija od značaja, za čiju je implementaciju kao ideja poslužio rad [43]. YOLO je najsavremeniji algoritam za detekciju objekata u realnom vremenu koji su 2015. godine predstavili Džozef Redmon i njegovi saradnici. Autori problem detekcije objekata predstavljaju kao regresiju, a ne kao zadatak klasifikacije i koriste jednu konvolucijsku neuronsku mrežu. Naziv algoritma se odnosi na činjenicu da zahteva samo jedan prolaz kroz neuronsku mrežu da bi napravio predviđanja, za razliku od prethodnih tehnika, što ga čini znatno bržim.

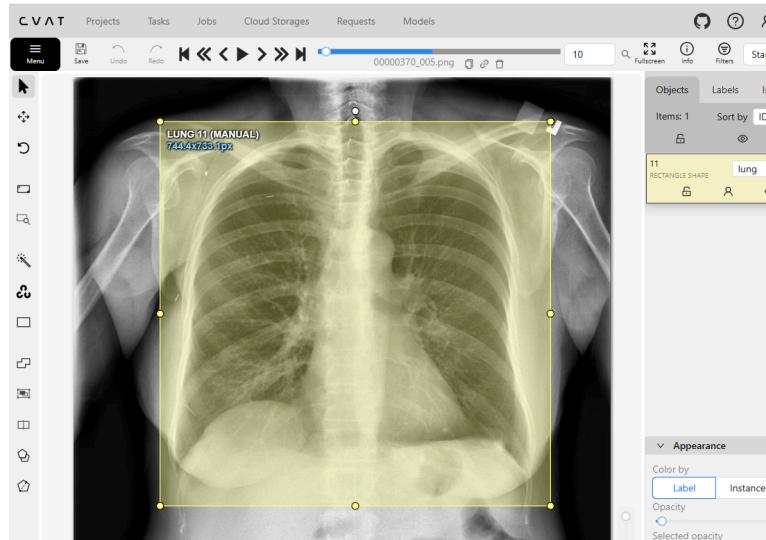
```

YOLO > yolo.py > ...
1  from ultralytics import YOLO
2
3  # Load a pretrained YOLOv8 model
4  model = YOLO("yolov8s.pt")
5
6  # Train the model
7  results = model.train(data="lung_det.yaml", epochs=50, imgsz=640, batch=2, workers=0, device=0,
8  | | | | | cache=False, project="lung_yolo_runs", name="lung_detector")
9
10 # Evaluate the model on validation set
11 metrics = model.val()
12 print("Validation metrics:", metrics)
13
14 # Test on a few images to visually inspect
15 model.predict(source="test_images/", save=True, imgsz=640, conf=0.25)
16
17 print("\nTraining complete!")
18 print("Best model saved at: lung_yolo_runs/yolo_lung_det/weights/best.pt")

```

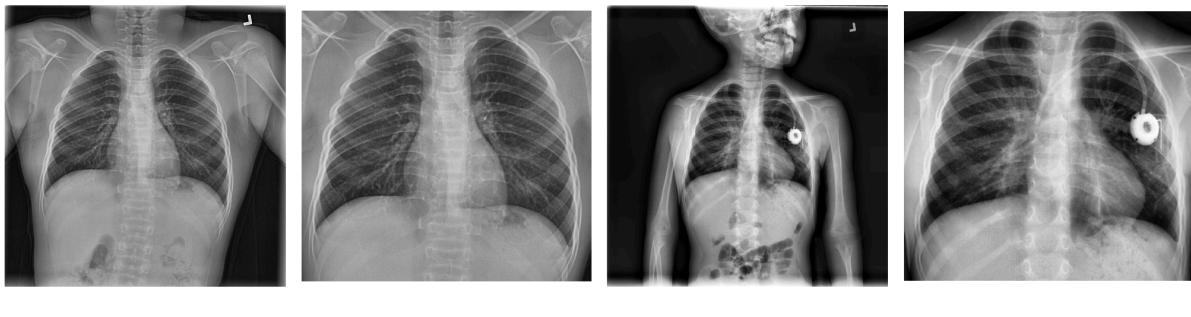
Slika XX: Treniranje YOLO algoritma za prepoznavanje plućnog polja

Tokom procesa pripreme podataka, YOLO model je obučen da automatski prepoznaće plućna krila na medicinskim slikama. Za njegovo treniranje izdvojeno je 375 slika, kojima su pridružene labele kreirane korišćenjem CVAT.ai, alata otvorenog koda za anotiranje slika. Proces ručnog anotiranja prikazan je na SlikaXX, nakon koga se anotacije mogu preuzeti u formatu namenjenom za treniranje YOLO algoritma. Za validaciju je izdvojeno 20% podataka i dobijeno je 100% tačnosti, a kod korišćenja za obuku prikazan je na SlikaXX.



Slika XX: Ručno anotiranje plućnog polja u CVAT.ai online alatu

Nakon treniranja YOLO algoritma, njegove težine su sačuvane radi dalje upotrebe. Prilikom učitavanja slika trening, validacionog i test skupa podataka, sačuvani YOLO algoritam je korišćen za predviđanje graničnih okvira (*engl. bounding boxes*) plućnih krila, na osnovu kojih je zatim vršeno isecanje slike i tako uklonjene sporedne strukture i vizuelna buka. Ovakav pristup obezbeđuje da se modeli usmeravaju isključivo na klinički značajne delove slike i sprečava učenje neželjenih obrazaca iz nerelevantnih regiona.



a

b

c

d

Slika XX: Primeri CXR slika pluća pre (a,c) i nakon (b,d) isecanja na osnovu YOLO-m predviđenih graničnih okvira

Za učitavanje slika iz direktorijuma na disk kreirane su tri metode “**build_dataframe**”, “**load_image**” i “**df_to_dataset**”. Prva metoda vrši kreiranje dataframe¹⁵ strukture na osnovu strukture direktorijuma tako što uzima putanju do slike iz svakog direktorijuma koji se nalazi u osnovnom i dodeljuje joj klasu (labelu) koja je naziv tog direktorijuma. Takođe, ova metoda vrši enkodiranje labela kako bi se njihove vrednosti pripremile za korišćenje u neuronskim mrežama i to u binarni vektor (*engl. one-hot encoding*¹⁶).

```
def build_dataframe(directory):
    filepaths = []
    labels = []

    for class_dir in directory.iterdir():
        if class_dir.is_dir():
            # directories names in train_and_val_dir are the class names
            label = class_dir.name
            for file in class_dir.glob(".*"):
                filepaths.append(str(file))
                labels.append(label)

    df = pd.DataFrame({"filepath": filepaths, "label": labels})
    df["label_index"] = df["label"].astype('category').cat.codes

    # converts each numerical label into a one-hot vector
    # eg. "[1, 0, 0]" for "covid19" class, "[0, 1, 0]" for "pneumonia" class and "[0, 0, 1]"
    df["label_onehot"] = df["label_index"].apply(
        lambda x: tf.keras.utils.to_categorical(x, num_classes=len(df["label"].unique())))

    return df, df["label"].unique()
```

Slika XX: Metoda za kreiranje dataframe strukture prema strukturi direktorijuma

Druga metoda je jako jednostavna i samo vrši učitavanje slika u odgovarajućoj veličini koja joj je prosleđena u vidu parametara željene visine i širine slike u pikselima.

Na Slika XX prikazana je poslednja metoda koja vrši kreiranje Tensorflow skupa podataka grupisanjem informacija sačuvanih u dataframe-u i slika koje se učitavaju pomoću metode “**load_image**”. Pored toga, poziva se i metoda “**prefetch**” kojom se podešava da CPU pripremi

¹⁵ **DataFrame** je dvodimenzionalna, promenljiva, tabelarna struktura podataka u Python-u.

¹⁶ **One-hot encoding** je tehnika za predstavljanje kategoričkih labela u binarne vektore, gde svaki vektor ima jedan element postavljen na 1, a ostatak na 0, što ukazuje na prisustvo ili odsustvo kategorije.

sledeću seriju podataka dok GPU trenutnu koristi za treniranje modela, što omogućava paralelno učitavanje i poboljšanje performansi obuke.

```
def df_to_dataset(df, shuffle=True):
    paths = df["filepath"].values
    labels = np.stack(df["label_onehot"].values)
    ds = tf.data.Dataset.from_tensor_slices((paths, labels))

    if shuffle:
        ds = ds.shuffle(len(df))

    # loading the images to dataset and resizing them
    ds = ds.map(lambda x, y: load_image(x, y), num_parallel_calls=AUTOTUNE)
    # prepare the next batch in the background while the model is training on the current
    ds = ds.batch(batch_size).prefetch(AUTOTUNE)
    return ds
```

Slika XX: Metoda za kreiranje skupa podataka

Deo preprocesiranja koji zavisi od CNN modela izabranog za transfer učenja izvršen je korišćenjem “keras.applications.preprocess_input” funkcije posebno dizajnirane za pripremu ulaznih slika za prethodno obučene modele dostupne u ”tensorflow.keras.applications”. Ovi modeli, kao što su VGG, ResNet, Inception i Xception, obučeni su na skupovima podataka kao što je ImageNet i očekuju određenu vrstu prethodne obrade za optimalne performanse.

```
preprocess_input = tf.keras.applications.mobilenet_v2.preprocess_input
# preprocessing becomes an actual Keras layer in the model graph
x = Lambda(preprocess_input, name="preprocessing_layer")(x)
```

Slika XX: Kreiranje preprocess_input sloja MobileNetV2 neuronske mreže

Svaki prethodno obučen model u ”tensorflow.keras.applications” ima svoju funkciju “preprocess_input”, npr. ”tensorflow.keras.applications.vgg16.preprocess_input”. Ova funkcija obrađuje ulazne slike kako bi poravnala sa formatom na kojem je određeni model obučen.

Te obrade uključuju:

- Skaliranje vrednosti piksela na određeni opseg (npr., -1 do 1, ili 0 do 1, ili normalizacija određene srednje vrednosti/standardne devijacije). Na ovaj način se vrednosti piksela (ovo je slučaj kada se koristi “rgb” format) iz opsega 0-255 prevode u opseg 0-1.
- Konvertovanje redosleda kanala boja između RGB i BGR, jer su neki modeli obučeni pomoću BGR slika.

Augmentacija podataka

Jedan od pristupa rešavanju problema pretreniranja mreže i problema male količine dostupnih podataka iskorišćenih u ovom radu je augmentacija podataka.

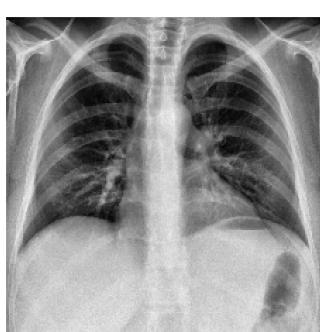
```

data_augmentation = tf.keras.Sequential([
    RandomFlip("horizontal"),
    RandomRotation(0.1),
    RandomZoom(height_factor=0.1, width_factor=0.1),
    RandomTranslation(height_factor=0.1, width_factor=0.1)
])

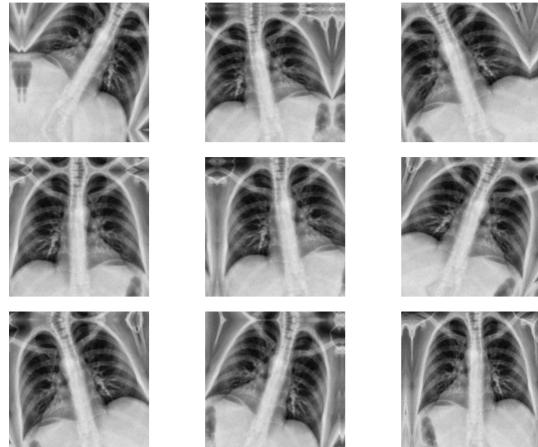
```

Slika XX: Implementacija objekta za augmentaciju podataka

Na SlikaXX prikazan je objekat za augmentaciju podataka koji koristi modul slojeva biblioteke Keras. Ovaj objekat obuhvata slučajnu rotaciju za faktor 0.1, što je 36 stepeni u smeru kazaljke na satu i u suprotnom smeru, zatim slučajnu translaciju vertikalno i horizontalno za faktor 0.1 što je 20ak piksela, zumiranje za faktor 0.1 što je 10% i horizontalno okretanje slike. Ovako definisani objekat radi tako što na mestu aktivacije vrši navedene transformacije, ali svaki put sa različitim parametrima da bi se dobilo na većoj varijabilnosti. Rezultati primene predstavljene augmentacije na jednoj od slika trening skupa, devet puta za redom, su prikazani na SlikaXX.



Slika XX: Slika pre vršenja augmentacije



Slika XX: Slike nakon augmentacije

Transfer učenja

U prethodnim poglavljima je detaljnije objašnjen pojam transfera učenja, kao i tri osnovne verzije ovog pristupa. Prilikom implementacije zadatka klasifikacije korišćene su dve metode i to plitko podešavanje ili kako se često naziva izdvajanje karakteristika i fino podešavanje.

Izdvajanje karakteristika

Najpre je za svaki korišćeni model učitavana konvolucionna osnova modela trenirana nad ImageNet skupom podataka iz biblioteke Keras, a zatim kreiran klasifikator takav da odgovara kako učitanoj mreži tako i konkretnom zadatku klasifikacije. Svi parametri konvolucione osnove

modela su zamrzavane, a treniran je samo novi klasifikator tako da se prilagodi konkretnom problemu. Nakon treniranja model je korišćenjem funkcije `save_feature_extraction_model` sačuvan u odgovarajućoj datoteci na disku ne bi li kasnije bio dostupan za nastavak obuke.

```
def build_model(num_classes):
    # data preprocessing
    inputs = Input(shape=(image_height, image_width, 3))
    x = data_augmentation(inputs)
    preprocess_input = tf.keras.applications.mobilenet_v2.preprocess_input
    # preprocessing becomes an actual Keras layer in the model graph
    x = Lambda(preprocess_input, name="preprocessing_layer")(x)

    # pretrained cnn
    base_model = MobileNetV2(input_tensor=x, include_top=False, weights='imagenet')
    # freezing the pretrained weights
    base_model.trainable = False

    # new classifier
    x=base_model.output
    x = GlobalAveragePooling2D()(x)
    x = Dense(128, activation="relu")(x)
    x = Dropout(0.5)(x)
    outputs = Dense(3, activation='softmax')(x)

    model = Model(inputs, outputs)

    metrics = [CategoricalAccuracy(name='accuracy'), Precision(name='precision'), Recall(name='recall')]
    model.compile(optimizer=Nadam(learning_rate=learning_rate), loss=CategoricalCrossentropy(), metrics=metrics)
    return model, base_model
```

Slika XX: Kreiranje modela CNN-a

Fino podešavanje

Prvi korak pri implementaciji finog podešavanja je učitavanje prethodno plitko podešenog istog modela konvolucijske neuronske mreže koji smo zapamtili u odgovarajućoj datoteci na disku. Takođe je potrebno odrediti za svaki model od kog sloja će se vršiti fino podešavanja, što se najčešće čini uzevši preporučene slojeve iz dokumentacije Keras biblioteke, a zatim se može i testiranjem utvrditi da li baš taj ili neki niži/viši sloj daju bolje rezultate u predikcijama za konkretni problem. Svakako je najvažnije da to bude tek nekoliko slojeva sa vrha mreže jer se radi o problemu male količine podataka te bi u suprotnom mrežu doveli do loših rezultata.

Drugi korak je odmrzavanje svih slojeva do odabranog sloja i zatim se vrši zajedničko treniranje klasifikatora i gornjih slojeva prethodno obučene mreže.

```

for fold, (train_idx, val_idx) in enumerate(kf.split(df)):
    print(f"\n===== FOLD {fold} =====\n")

    train_df = df.iloc[train_idx]
    val_df = df.iloc[val_idx]

    train_ds = df_to_dataset(train_df, shuffle=True)
    val_ds = df_to_dataset(val_df, shuffle=False)

    print("Train samples:", len(train_idx))
    print("Val samples:", len(val_idx))

    model, base_model = build_model(num_classes=len(class_names))

    history = model.fit(train_ds, epochs=epochs, validation_data=val_ds, callbacks=callbacks)

    # Fine-tuning last 10 layers
    for layer in base_model.layers[-10:]:
        layer.trainable = True

    for layer in base_model.layers:
        if isinstance(layer, tf.keras.layers.BatchNormalization):
            layer.trainable = False

    fine_tuning_learning_rate = 5e-5
    model.compile(loss=CategoricalCrossentropy(),
                  optimizer=Adam(learning_rate=fine_tuning_learning_rate),
                  metrics=[CategoricalAccuracy(name='accuracy')])

    total_epochs = epochs + fine_tune_epochs
    history_fine = model.fit(train_ds, epochs=total_epochs, initial_epoch=len(history.epoch), validation_data=val_ds, callbacks=callbacks)

    print(f"✓ Finished Fold {fold+1}\n")

    val_accuracy_list.append(history.history["val_accuracy"][-1])
    recall_list.append(history.history["recall"][-1])
    precision_list.append(history.history["precision"][-1])

    # save acc and loss plots
    save_fold_plots(history, fold)

print("\n===== CROSS-VAL RESULTS =====\n")
print("Fold accuracies:", val_accuracy_list)
print("Mean accuracy:", np.mean(val_accuracy_list))
print("Fold recalls:", recall_list)
print("Mean recall:", np.mean(recall_list))
print("Fold precisions:", precision_list)
print("Mean precision:", np.mean(precision_list))

```

Slika XX: Implementacija 5-ostrukne unakrsne validacije

Rezultati istraživanja

Zaključak

Literatura

- [1] C.D.C. COVID, R. Team (February 12 - March 16, 2020) ‘Severe outcomes among patients with coronavirus disease 2019 (COVID-19)—United States’, *MMWR Morb Mortal Wkly Rep.* 69, 343–346.
- [2] A. Remuzzi, G. Remuzzi (March 13, 2020) ‘COVID-19 and Italy: what next?’, *The Lancet*, 395(10231), 1225-1228.
- [3] Joe Parkin Daniels (July 25, 2020) ‘COVID-19 cases surge in Colombia’, *The Lancet*, 396(10246), 227.
- [4] Krutika Kuppalli, Pooja Gala, Kartikeya Cherabuddi, S P Kalantri, Manoj Mohanan, Bhramar Mukherjee et al. (May 14, 2021) ‘India’s COVID-19 crisis: a call for international action’, *The Lancet*, 397(10290), 2132-2135.
- [5] E. Mahase (2020) ‘Coronavirus: covid-19 has killed more people than SARS and MERS combined, despite lower case fatality rate’, *BMJ*, 368:m641.
- [6] Nikhil Ketkar (2017) ‘Deep Learning with Python: A Hands-on Introduction’, 1st edn. United States of America: Apress Berkeley CA.
- [7] Josh Patterson, Adam Gibson (2017) ‘Deep Learning: A practitioner’s approach’, 1st edn. United States of America: O'Reilly Media.
- [8] World Health Organization. (May 5. 2023) ‘WHO Director-General’s opening remarks at the media briefing - 5 May 2023’, Available at: <https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing---5-may-2023> (Accessed 28. October 2025)
- [9] V.M. Corman, O. Landt, M. Kaiser, R. Molenkamp et al. (2020) ‘Detection of 2019 novel coronavirus (2019-nCoV) by real-time RT-PCR’, *Eurosurveillance*, 25 2000045.
- [10] T. Ai, Z. Yang, H. Hou, C. Zhan et al. (Feb 26, 2020) ‘Correlation of Chest CT and RT-PCR Testing for Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases’, *RSNA*. 296(2), E32–E40.
- [11] M.Y. Ng, E. YP Lee, J. Yang, F. Yang, X. Li, H. Wang et al. (Feb 13 2020) ‘Imaging profile of the COVID-19 infection: Radiologic findings and literature review’, *Cardiothoracic Imaging*, 2(1).
- [12] C. Huang Y. Wang et al. (2020) ‘Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China’, *The Lancet*, 395, 497–506.
- [13] W. Guan, Z. Ni, Y. Hu, W. Liang, C. Ou, J. He et al. (2020) ‘Clinical characteristics of coronavirus disease 2019 in China’, *N Engl J Med*, 382, 1708-1720
- [14] A. Krizhevsky, I. Sutskever, and G. Hinton (2012) ‘ImageNet classification with deep convolutional networks’, *Advances in Neural Processing Systems*, 25, 1097–1105.

- [15] Neha Patha (September 23, 2020) ‘What Does COVID-19 Do to Your Lungs?’, *WebMD*. Available at: <https://www.webmd.com/lung/what-does-covid-do-to-your-lungs#1> (Accessed 16 September 2025)
- [16] C. Rachna (April 5, 2017) ‘Difference between X-ray and CT Scan’, Bio Differences. Available at: <https://biodifferences.com/difference-between-x-ray-and-ct-scan.html> (Accessed 16 September 2025)
- [17] J.P. Cohen, P. Morrison, L. Dao (2020) ‘COVID-19 image data collection’, *ArXiv Prepr*: ArXiv2003.11597.
- [18] A. Narin, C. Kaya, Z. Pamuk (2021) ‘Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks’, *Pattern Anal Applic* 24, 1207-1220.
- [19] I.D. Apostolopoulos, T.A. Mpesiana (2020) ‘Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks’, *Phys Eng Sci Med* 43, 635–640.
- [20] L. Wang, Z. Lin, A. Wong (2020) ‘Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest radiography images’, *ArXiv Prepr*: ArXiv2003.09871.
- [21] E.E.-D. Hemdan, M.A. Shouman, M.E. Karar (2020) ‘Covidx-net: A framework of deep learning classifiers to diagnose covid19 in x-ray images’, *ArXiv Prepr*: ArXiv2003.11055.
- [22] François Chollet (2018) ‘Deep Learning with Python’, 1st edn. United States of America: Manning Publications Co.
- [23] Aurélien Géron (2017) ‘Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems’, 1st edn. United States of America: O'Reilly Media.
- [24] Muhammad Imran Razzak, Saeeda Naz and Ahmad Zaib (2018) ‘Deep Learning for Medical Image Processing: Overview, Challenges and Future’, *Classification in BioApps. Lecture Notes in Computational Vision and Biomechanics*, vol 26. Springer, Cham.
- [25] Sindhu, Vaidhehi V. (2017) ‘Classification of Human Organ Using Image Processing’, *Orient.J. Comp. Sci. and Tech*, 10(2).
- [26] D. Hubel and T. Wiesel (1958) ‘Single Unit Activity in Striate Cortex of Unrestrained Cats’, *The Journal of physiology*, 147(2), 226–238.
- [27] D. Hubel and T. Wiesel (1959) ‘Receptive Fields of Single Neurons in the Cat’s Striate Cortex’, *The Journal of physiology*, 148(3), 574–591.
- [28] D. Hubel and T. Wiesel (1968) ‘Receptive Fields and Functional Architecture of Monkey Striate Cortex’, *The Journal of physiology*, 195, 215–243.
- [29] Y. LeCun et al. (1998) ‘Gradient-Based Learning Applied to Document Recognition’, *Proceedings of the IEEE*, 86(11), 2278-2324.
- [30] J.M. Johnson, T.M. Khoshgoftaar (2019) ‘Survey on deep learning with class imbalance’ *J Big Data* 6, 27.

- [31] L. Lu, Y. Zheng, G. Carneiro, and L. Yang (2017) ‘Deep Learning and Convolutional Neural Networks for Medical Image Computing: Precision Medicine, High Performance and Large-Scale Datasets’, *Springer*.
- [32] G. Mohan, M.M. Subashini, S. Balan *et al.* (2024) ‘A multiclass deep learning algorithm for healthy lung, Covid-19 and pneumonia disease detection from chest X-ray images’, *Discov Artif Intell* 4, 20.
- [33] I. Banerjee, P. Sinha, S. Purkayastha, N. Mashhaditafreshi, A. Tariq, J. Jeong, H. Trivedi, J. W. Gichoya (2020) ‘Was there COVID-19 back in 2012?’, *Challenge for AI in Diagnosis with Similar Indications*, ArXiv2006.13262.
- [34] M. Roberts, D. Driggs, M. Thorpe *et al.* (2021) ‘Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans.’ *Nat Mach Intell.* 3, 199–217.
- [35] M. Chakraborty, S.V. Dahvale, J. Ingole (2021) ‘Corona-Nidaan: lightweight deep convolutional neural network for chest X-Ray based COVID-19 infection detection’, *Appl Intell* 51, 3026-3043.
- [36] M. Rahimzadeh, A. Attar (2020) ‘A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-Ray images based on concatenation of Xception and ResNet50V2’, *Informatics in Medicine*, 19, 2352-9148
- [37] A. Abbas, M.M. Abdelsamea, M.M. Gaber (2021) ‘Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network’, *Appl Intell* 51, 854-864.
- [38] P. An, H. Chen, X. Jiang, J. Su, Y. Xiao, Y. Ding, H. Ren, M. Ji, Y. Chen, W. Chen *et al.* (February 6, 2020) ‘Clinical features of 2019 novel coronavirus pneumonia presented gastrointestinal symptoms but without fever onset’, *SSRN*, 3532530: (Preprint).
- [39] JF-W. Chan, S. Yuan, K-H Kok *et al.* (February 15, 2020) ‘A familial cluster of pneumonia associated with the 2019 novel coronavirus indicating person-to-person transmission: a study of a family cluster’, *The Lancet*, 395, 514–523.
- [40] T. Sanida, M. Dasygenis (2024) ‘A novel lightweight CNN for chest X-ray-based lung disease identification on heterogeneous embedded system’, *Appl Intell* 54, 4756-4780.
- [41] N.S. Punn, S. Agarwal (October 17, 2020) ‘Automated diagnosis of COVID-19 with limited posteroanterior chest X-ray images using fine-tuned deep neural networks’, *Appl Intell* 51, 2689-2702
- [42] S. Hira, A. Bai, S. Hira (November 27. 2020) ‘An automatic approach based on CNN architecture to detect COVID-19 disease from chest X-ray images’, *Appl Intell* 51, 2864-2889
- [43] A. Karaci (January 24. 2022) ‘VGGCOV19-NET: automatic detection of COVID-19 cases from X-ray images using modified VGG19 CNN architecture and YOLO algorithm’, *Neural Comput & Applic* 34, 8253-8274
- [44] R.D. Bhosale, D.M. Yadav (2024) ‘Customized convolutional neural network for pulmonary multi-disease classification using chest x-ray images’, *Multimed Tools Appl* 83, 18537–18571.
- [45] T. Zebin, S. Rezvy (September 12. 2020) ‘COVID-19 detection and disease progression visualization: Deep learning on chest X-rays for classification and coarse localization’, *Appl Intell* 51, 1010-1021

- [46] A. Paul, A. Basu, M. Mahmud et al (January 5. 2022) ‘Inverted bell-curve-based ensemble of deep learning models for detection of COVID-19 from chest X-rays’, *Neural Comput & Applic* (2022)
- [47] J. Zhang, Y. Xie, Y. Li, C. Shen, Y. Xia (March 27. 2020) ‘Covid-19 screening on chest x-ray images using deep learning based anomaly detection’, *arXiv:2003.12338*
- [48] H. Gupta, N. Bansal, S. Garg, H. Mallik, A. Prabha, J. Yadav (2023) ‘A hybrid convolutional neural network model to detect covid-19 and pneumonia using chest x-ray images’, *Int J Imaging Syst Technol* 33(1):39–52.
- [49] T. Ozturk, M. Talo, E.A. Yildirim, U.B. Baloglu, O. Yildirim, U.R. Acharya (June 2020) ‘Automated detection of COVID-19 cases using deep neural networks with X-ray images’, *Computers in Biology and Medicine* 121:103792
- [50] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, RM. Summers (2017) ‘ChestX-Ray8: hospital-scale chest X-ray database om weakly-supervised classification and localization of common thorax diseases’, *IEEE conference on computer vision and pattern recognition (CVPR)* pp 3462-3471.
- [51] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio (2014) ‘Generative adversarial networks’, *arXiv preprint arXiv:1406.2661*.
- [52] B. K. Umri, E. Utami, M. P. Kurniawan (2021) ‘Comparative Analysis of CLAHE and AHE on Application of CNN Algorithm in the Detection of Covid-19 Patients’, *4th International Conference of Information and Communication Technology (ICOIACT)* pp 203-208.
- [53] Z. He, A.B. McMillan (2025) ‘Comparative Evaluation of Radiomics and Deep Learning Models for Disease Detection in Chest Radiography’, *J Digit Imaging. Inform. Med.*
- [54] François Chollet (2015) ‘Keras: the Python deep learning API’ Available at: <https://keras.io/> (Accessed 5 November 2025)
- [55] V. Sampath, I. Maurtua, J.J. Aguilar Martín, et al. (2021) ‘A survey on generative adversarial networks for imbalance problems in computer vision tasks’, *J Big Data* 8, 27.
- [56] T-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár (2017) ‘Focal loss for dense object detection’, *Proceedings of the IEEE International conference on computer vision*, pp 2980–2988.
- [57] M. Gholampour (2024) ‘Impact of nature of medical data on machine and deep learning for imbalanced datasets: Clinical validity of SMOTE is questionable’, *Machine Learning and Knowledge Extraction*, 6(2), 826–836.
- [58] A. M. Tahir, M. E. H. Chowdhury, A. Khandakar, et al (2021) ‘COVID-19 Infection Localization and Severity Grading from Chest X-ray Images’, *Computers in Biology and Medicine* 139:105002.
- [59] A. M. Tahir, M. E. H. Chowdhury, Y. Qiblawey, et al (2021) ‘COVID-QU-Ex’, *Kaggle*.
- [60] T. Rahman, A. Khandakar, Y. Qiblawey A. Tahir, et al (2021) ‘Exploring the Effect of Image Enhancement Techniques on COVID-19 Detection using Chest X-rays Images‘ *Computers in Biology and Medicine*, 104319.

[61] A. Degerli, M. Ahishali, M. Yamac, S. Kiranyaz, et al (2021) ‘Covid-19 infection map generation and detection from chest X-ray images‘ *Health Inf Sci Syst* 9, 15.

[62] M. E. H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, et al (2020) ‘Can AI Help in Screening Viral and COVID-19 Pneumonia?‘, *IEEE Access* 8, 132665-132676.