

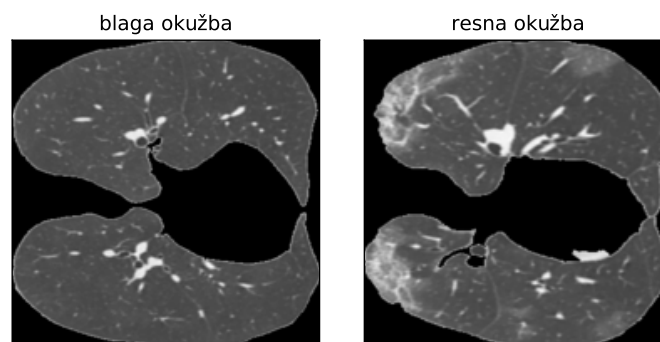
Binarna klasifikacija resnosti okužbe COVID-19 z analizo CT-slik

Praktikum strojnega učenja v fiziki

Domen Vaupotič

1 COVID-19

V nalogi bomo obravnavali medicinske slike prsnega koša, posnete z računalniško tomografijo (CT) pri bolnikih, ki so prebolevali COVID-19. Na sliki 1 sta prikazani rezina pljuč pri bolniku z blagim in pri bolniku z resnim potekom okužbe. S strojnim učenjem želimo ustvariti klasifikator, ki bi bil sposoben avtonomno in nedvoumno diagnosticirati posnetek CT in ovrednotiti resnost bolnikovega stanja. Zlati standard določanja okužbe še zmeraj predstavlja meritev prisotnosti virusnega genoma, ki je v splošnem bolj specifična in občutljiva metoda, prav tako pa cenejša in za bolnika mnogo manj škodljiva kot preiskava CT. Diagnostika s slikanjem CT je torej smiselna le v pogojih, ko meritev prisotnosti virusnega genoma ni mogoča, ali pa kot diagnostično-prognostična ocena bolnikovega stanja. Pokazalo se je namreč, da lahko spremembe na pljučih napovedujejo resnejši potek bolezni, še preden se pojavijo drugi značilni znaki in simptomi. Ker je za interpretacijo rezultatov CT potreben specialist radiologije, bi s strojnim učenjem tako omogočili predvsem hitrejše pregledovanje slik. Če je slikanje CT razmeroma dolgotrajno, zahteva drago medicinsko opremo in je za bolnika razmeroma škodljivo, pa bi še idealnejša možnost bila diagnostika na podlagi rentgenskih slik prsnega koša.



Slika 1: Primer slike CT pri bolniku z blago okužbo, kjer ni vidnih večjih sprememb pljučnega tkiva (levo), in pri bolniku z resno okužbo, pri katerem se pojavijo značilni vzorci mlečnega stekla (GGO) in obrnjeni znak halo (desno).

2 Nabor podatkov

Podatki v nalogi so del prostodostopne zbirke MOSMED, v kateri so anonimizirane CT-slike prsnega koša, zbrane med marcem in aprilom 2020. Slike so velikosti 512×512

pikslov. Izbrani so podatki za 226 pacientov z blago okužbo in 226 pacientov z resno okužbo COVID-19. Strojno učenje je potekalo na 272 učnih slikah (60 %), ki predstavljajo centralne rezine. Za validacijo med učenjem je bilo na voljo $90 \times 10 = 900$ slik (20 %), ki pripadajo 90 bolnikom, za vsakega 10 rezin. Modele smo ovrednotili na dodatnih neodvisnih 900 slikah (20 %), ponovno 90 bolnikov z 10 rezinami. Končni rezultat za vsakega bolnika je povprečje rezultatov vseh 10 rezin. Dodatno testiranje mrež smo izvedli tudi na lastnih slikah CT.

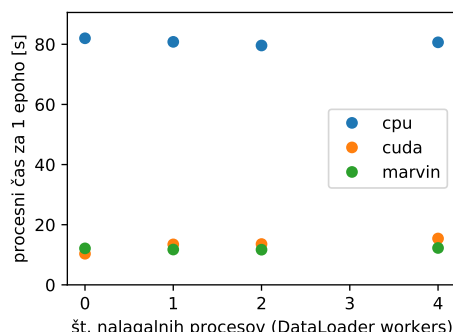
3 Rezultati

3.1 Časovna zahtevnost

Ker je nabor podatkov razmeroma velik, je treba previdno izbrati računsko enoto. Za izbiro sem naredil testno učenje nevronske mreže ResNet18 na treh računskih enotah:

- ★ **CPU** – procesor prenosnega računalnika
Intel Core i5-8250U @ 1,60 GHz, 4 jedra, 8 logičnih procesorjev, 8 GB RAM
- ★ **CUDA** – grafična kartica prenosnega računalnika
Nvidia GeForce MX150 @ 1,50GHz, 384 jeder CUDA, 2 GB RAM
- ★ **Marvin** – procesor strežnika Marvin
Intel Xeon Gold 6248R @ 3,00GHz, 20 jeder, 40 logičnih procesorjev, 76 GB RAM

Zaradi majhnega spomina grafične kartice in znatne velikosti mreže ResNet18 je bila največja možna velikost učnega paketa (*batch size*) 9 slik, zato sem pri testiranju časovne zahtevnosti pri vseh računskih enotah uporabil takšno velikost. Omogočena je bila nastavek `pin_memory`, ki omogoča hitrejši prenos podatkov med CPU in GPU. Zanimalo me je, ali parameter `num_workers` objekta `DataLoader`, ki podaja število niti, s katerimi nalagamo podatke, vpliva na računski čas. S tem bi lahko ugotovili, ali se pojavlja ozko grlo pri pridobivanju podatkov z diska. Na sliki 2 so prikazani računski časi za učenje mreže ResNet18 na treh različnih računskih enotah. Vidimo, da se procesor obnese mnogo slabše kot grafična kartica in strežnik Marvin, ki sta med seboj primerljiva. Število nalagalnih procesov nima bistvenega vpliva na procesni čas, zato sem v vseh nadaljnjih izračunih uporabljal 2 nalagalna procesa.



Slika 2: Primerjava računskega časa za tri različne računske enote.

Za učenje nevronske mreže sem v nadaljevanju uporabil strežnik Marvin, za testiranje in vso nadaljnjo uporabo mreže pa sem uporabljal grafično kartico v svojem računalniku.

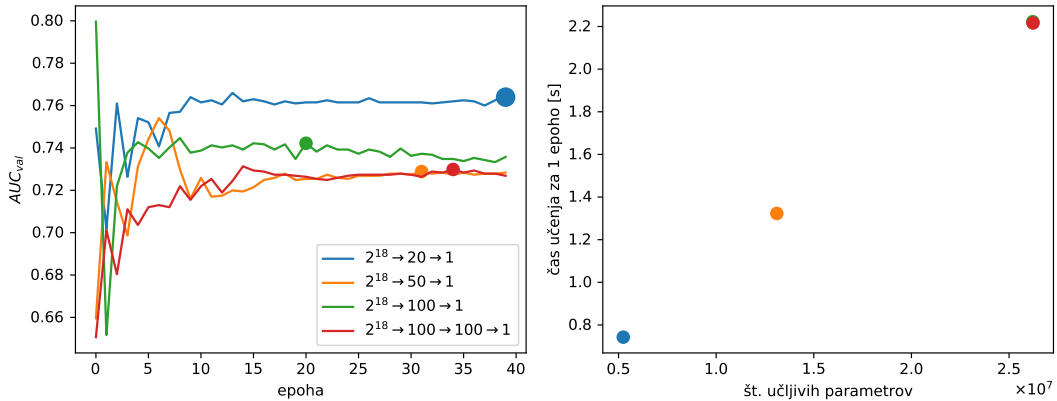
Omeniti še velja, da okolje PyTorch omogoča enostavno preklapljanje med računskimi enotami, vendar je uporaba grafične kartice včasih nestabilna in povzroča nepredvidene zastoje programa ali prekoračitve spomina.

3.2 Navadna gosto povezana mreža

Najenostavnejši pristop, ki ga lahko uberemo, je navadna gosto povezana usmerjena nevronska mreža. Pri tej vhodno sliko pretvorimo v vektor dolžine 2^{18} , ki ga vodimo skozi nekaj zaporednih plasti popolnoma povezanih nevronov. Ker je velikost vhodnih podatkov zelo velika, je tudi velikost mreže že pri majhnem številu nevronov v naslednjih plasteh zelo velika, s tem pa znatno naraste tudi število učljivih parametrov. Preveril sem štiri različne arhitekture, in sicer z eno ali dvema skritima plastema:

- ★ $2^{18} \rightarrow 20 \rightarrow 1$
- ★ $2^{18} \rightarrow 50 \rightarrow 1$
- ★ $2^{18} \rightarrow 100 \rightarrow 1$
- ★ $2^{18} \rightarrow 100 \rightarrow 100 \rightarrow 1$

Med skritima plastema je bila uporabljena aktivacijska funkcija ReLU, za pridobitev končnega rezultata pa je uporabljena sigmoidna aktivacijska funkcija. Na sliki 3 so prikazani poteki učenja za navadne nevronske mreže in število učljivih parametrov v primerjavi z računskim časom. Opazimo, da s številom parametrov narašča tudi čas učenja. Zanimivo pa je, da se je najbolje obnesla najenostavnejša mreža, torej mreža z zgolj eno skrito plastjo 20 nevronov, čeprav bi pričakovali, da bo kakovost mreže naraščala z njeno kompleksnostjo. Vzrok za tak rezultat je verjetno v premajhni izbrani hitrosti učenja za večje mreže. Uporabljen validacijski parameter je AUC (*area under curve*), ki bo podrobneje opisan v nadaljevanju. Za najboljšo navadno mrežo sem izbral mrežo z največjim AUC.



Slika 3: Primerjava poteka učenja pri navadnih globokih nevronske mrežah (levo) ter odvisnost učnega časa od števila prostih učljivih parametrov (desno). Najboljši rezultat je prikazan z večjo piko. Hitrost učenja $\beta = 5 \times 10^{-3}$ z zmanjševalnim faktorjem 0,95, stopnja regularizacije $wd = 10^{-4}$, velikost učnega paketa 15.

3.3 Konvolucijska nevronska mreža

Ker so vhodni podatki slike, je smiselnejša izbira konvolucijskih nevronskih mrež, ki so boljše za prepoznavanje vzorcev v slikah. Sestavil sem dve različni arhitekturi, in sicer:

★ CNN1

```
– Conv2D(1, 5, kernel=(5, 5), stride=(3, 3), padding=(3, 3))
– BatchNorm2D(5)
– relu
– MaxPool2D
– Conv2D(5, 5, kernel=(5, 5), stride=(2, 2), padding=(3, 3))
– BatchNorm2d(5)
– RelU
– MaxPool2D
– Conv2D(5, 5, kernel=(5, 5), stride=(2, 2), padding=(3, 3))
– BatchNorm2D(5)
– RelU
– MaxPool2D
– Linear(180, 180)
– RelU
– Linear(180, 1)
```

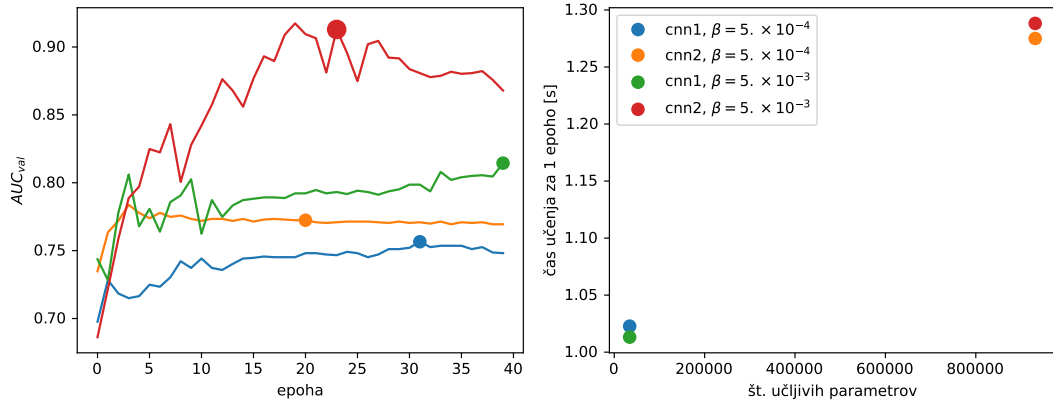
★ CNN2

```
– Conv2D(1, 10, kernel=(5, 5), stride=(3, 3), padding=(3, 3))
– BatchNorm2D(10)
– relu
– MaxPool2D
– Conv2D(10, 20, kernel=(5, 5), stride=(2, 2), padding=(3, 3))
– BatchNorm2d(20)
– RelU
– MaxPool2D
– Conv2D(20, 50, kernel=(5, 5), stride=(2, 2), padding=(3, 3))
– BatchNorm2D(50)
– RelU
– MaxPool2D
– Linear(1800, 500)
– RelU
– Linear(500, 1)
```

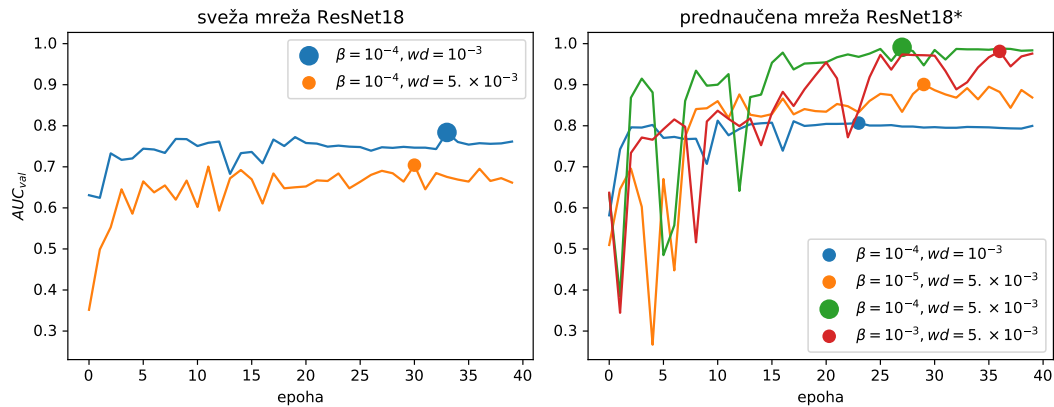
Končni rezultat je ponovno izračunan s sigmoidno aktivacijsko funkcijo na zadnjem nevronu. Pri sestavljanju druge konvolucijske mreže sem upošteval priporočila, da se število filtrov v plasteh z globino povečuje. Primerjava potekov učenja je prikazana na sliki 4. Bolj se je izkazala druga konvolucijska mreža, in sicer pri hitrejši stopnji učenja.

3.4 ResNet18

Tretja arhitektura nevronske mreže, ki sem jo preizkusil, je konvolucijska nevronska mreža ResNet18, ki je bila izdelana za prepoznavanje 1000 različnih razredov slik iz zbirke ImageNet. Preveril sem uspešnost učenja sveže mreže (ResNet18) in uspešnost učenja mreže, ki je bila predhodno naučena na zbirki ImageNet (ResNet18*). Obe mreži sta nekoliko prilagojeni, tako da kot vhodni podatek sprejmeta sivinsko sliko (in ne barvne), prav tako pa je odstranjena končna gosto povezana mreža in nadomeščena z združevalnikom MaxPool. Rezultati učenja so prikazani na sliki 5. Opazimo, da se prednaučena mreža obnese znatno bolje. Prav tako opazimo, da hitrost učenja močno vpliva na uspešnost mreže. Z dodatnim prilagajanjem hiperparametrov bi verjetno iz mreže uspeli iztisniti še kakšno odstotno točko več.



Slika 4: Primerjava poteka učenja pri navadnih globokih nevronske mrežah (levo) ter odvisnost učnega časa od števila prostih učljivih parametrov (desno). Najboljši rezultat je prikazan z večjo piko. Hitrost učenja $\beta = 5 \times 10^{-3}$ z zmanjševalnim faktorjem 0,95, stopnja regularizacije $wd = 1 \times 10^{-4}$, velikost učnega paketa 15.



Slika 5: Primerjava poteka učenja pri sveži mreži ResNet18 in pri mreži, ki je bila prednaučena na zbirki ImageNet.

3.5 Primerjava velikosti arhitektur

Pomudimo se na kratko še pri spominski požrešnosti posameznih arhitektur. Knjižnica `torchsummary` nam omogoča izračun velikosti spomina, ki ga zaseda nevronska mreža, kar je prikazano v tabeli 1. Opazimo veliko porabo spomina pri mreži ResNet18, ki je potrebna v času učenja (velikost prehoda), zaradi česar smo omejeni z velikostjo učnega paketa. Pri validaciji se velikost prehoda ne upošteva, saj ne izvajamo popravljanja uteži.

4 Binarna klasifikacija

Kadar se ukvarjamo z binarnim klasifikatorjem (pa najsibo to nevronska mreža ali laboratorijski diagnostični test), naletimo na težavo, kako ovrednotiti njegovo uspešnost. Surovi rezultat klasifikatorja je namreč skoraj zmeraj (implicitno ali eksplicitno) neko realno število, za katerega moramo postaviti ustrezni prag, da ga pretvorimo v končni binarni rezultat. Intuitivno je logično, da izbira vrednosti praga vpliva na uspešnost binarnega klasifikatorja.

	število učljivih param.	velikost parametrov	velikost prehoda
Dense(20)	5 242 921	20,00 MB	168 B
Dense(50)	13 107 301	50,00 MB	408 B
Dense(100)	26 214 601	100,00 MB	808 B
Dense(100, 100)	26 224 701	100,04 MB	1608 B
CNN1	34 166	0,13 MB	2,42 MB
CNN2	931 411	3,55 MB	5,22 MB
ResNet18	11 170 753	42,61 MB	328,00 MB

Tabela 1: Število učljivih parametrov in poraba spomina pri različnih arhitekturah.

Zgodbo začnemo pri definiranju možnih izidov. Naj bo rezultat binarnega klasifikatorja pozitiven izid (P) ali negativen izid (N). Če klasifikator pravilno napove pozitiven izid, je to *resnično pozitiven* izid (TP), če pravilno napove negativen izid, je to *resnično negativen* izid (TN). Klasifikator lahko naredi tudi dva tipa napake: prva je *lažno pozitivni* izid (FP), druga *lažno negativen* izid (FN). Z izbiro vrednosti praga vplivamo na razmerje deležev med pravilnimi in nepravilnimi klasifikacijami. Skoraj vedno je treba najti kompromis med lažno negativnimi in lažno pozitivnimi izidi. Okoliščine in namembnost klasifikatorja odločijo o tem, katerih želimo manj.

4.1 Mere uspešnosti

Uspešnost nekega binarnega klasifikatorja pri izbrani vrednosti praga v popolnosti predstavimo z matriko zamenjav, v kateri podamo številčne (ali odstotne) vrednosti za vse štiri možne izide. Ker pa nimamo pravega občutka, kako bi lahko med seboj primerjali dve matriki zamenjav različnih klasifikatorjev, običajno uvedemo dodatne količine, s katerimi delno popišemo matriko zamenjav:

$$\begin{aligned}
TPR &= \frac{TP}{TP+FN} && \text{občutljivost, senzitivnost, priklic} \\
TNR &= \frac{TN}{TN+FP} && \text{specifičnost, selektivnost} \\
FPR &= \frac{FP}{FP+TN} && \text{lažni alarm} \\
FNR &= \frac{FN}{FN+TP} && \text{zgrešeni pozitivni} \\
PPV &= \frac{TP}{TP+FP} && \text{natančnost} \\
ACC &= \frac{TP+TN}{TP+TN+FP+FN} && \text{točnost}
\end{aligned}$$

Opazimo, da vsaka izmed količin podaja neko razmerje znotraj matrike zamenjav. Količini, ki sta nam pogosto pomembni (posebej pri medicinskih testih), sta občutljivost (ta pove, kako dobro zna klasifikator najti pozitivne primere) in specifičnost (ta pove, kako dobro najdemo negativne primere). Ti dve količini lahko na različne načine ponovno združimo v enotno mero uspešnosti:

$$\begin{aligned}
BA &= \frac{1}{2} (TPR + PPV) && \text{uravnotežena točnost} \\
FM &= \sqrt{TPR \times PPV} && \text{indeks Fowlkes-Mallows} \\
F_1 &= 2 \frac{TPR \times PPV}{TPR + PPV} && \text{vrednost } F_1
\end{aligned}$$

Opazimo, da smo izračunali zgolj tri različne sredine: aritmetično, geometrično in harmonično. Omenimo še eno izmed mero uspešnosti, to je:

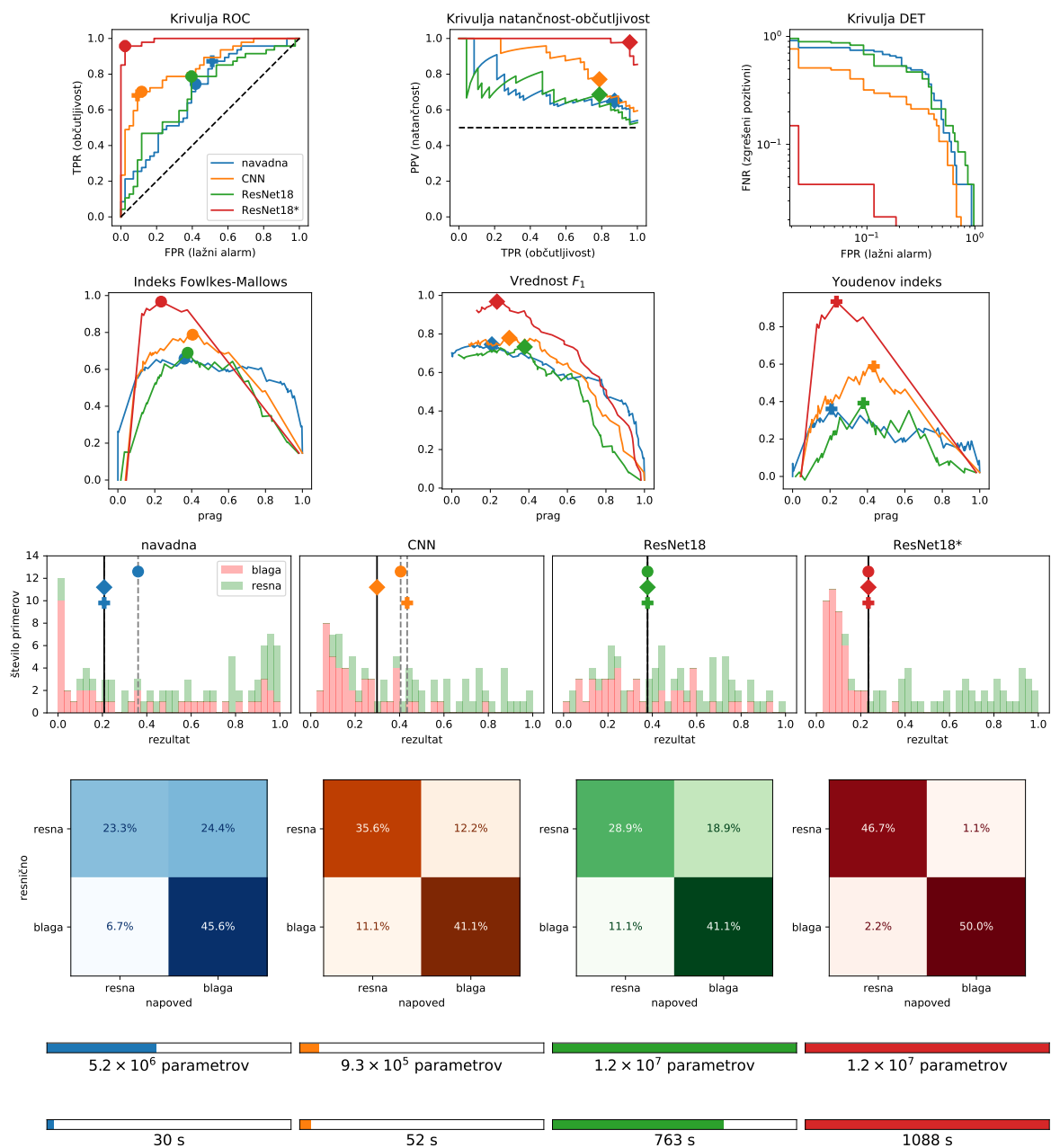
$$J = TPR + TNR - 1 \quad \text{Youdenov indeks}$$

Izkaže se, da izbiro ustrezne mere uspešnosti pogosto oteži še neenakomerna porazdelitev (resnično) pozitivnih in (resnično) negativnih primerov v opazovani populaciji, torej pojavnost (prevalenca). Prav zaradi nizke pojavnosti redkih bolezni je verjetnost, da ob pozitivnem testu bolezen tudi dejansko imamo, pravzaprav zelo majhna, navkljub visoki občutljivosti (in specifičnosti) diagnostičnega testa. V primerih neuravnotežene pojavnosti se zato pogosto uporablja tudi Matthewsov korelacijski koeficient (MCC), ki bolje odraža uspešnost klasifikatorja. Naj še enkrat poudarim, da so zgornje številke zgolj nepopolna reprezentacija matrike zamenjav, ki v polnosti podaja kakovost klasifikatorja.

Pri snovanju klasifikatorja pa torej še vedno naletimo na težavo, kako izbrati ustrezní prag. Po zgornji razpravi si torej lahko predstavljamo, da bomo prag izbrali tako, da bomo *maksimizirali* eno izmed (želenih!) mer za uspešnost klasifikatorja, pa najsibo to občutljivost, vrednost F_1 ali MCC. Prikladni so zato grafi, kjer hkrati predstavimo dve izmed količin iz matrike zamenjav pri vseh možnih izbranih vrednostih praga. Na ta način vidimo, kakšen kompromis bomo morali narediti med njima. Tipične krivulje so tako krivulja ROC (*receiver operating characteristic*), ki prikazuje TPR in FPR, krivulja natančnosti občutljivosti ter krivulja DET (*detection error tradeoff*), ki prikazuje FPR in FNR. S krivuljo ROC pridobimo še eno zanimivo mero uspešnosti klasifikatorja, to je vrednost AUC (*area under ROC curve*), ki podaja ploščino pod grafom ROC. Vrednost AUC nam pove verjetnost, da bo izmed naključno izbranega (resnično) pozitivnega primera in naključno izbranega (resnično) negativnega primera iz populacije podelil klasifikator pozitivnemu primeru *večji* končni rezultat. Intuitivno nam torej višji AUC pove večjo/boljšo ločljivost pozitivnih in negativnih primerov. Vrednost AUC ima torej (za razliko od prej omenjenih mer uspešnost) dobro lastnost, da je neodvisna od izbrane vrednosti praga, izkaže pa se, da je tudi neodvisna od razlik v pojavnosti.

4.2 Primerjava rezultatov

Oglejmo si sedaj uspešnost najboljših štirih nevronske mreže. Zbrani rezultati so prikazani na slikah 6 in 7. Odločil sem se, da za prag določim z maksimizacijo vrednosti F_1 , saj sem po pregledu histograma videl, da je bil ta prag vedno najnižji izmed vseh treh, s čimer sem zmanjšal število lažno negativnih izidov, kar nam je pri medicinski diagnostiki skoraj zmeraj pomembnejše. Ne glede na izbrano mero uspešnosti vidimo, da nevronska mreža ResNet18*, ki je bila prednaučena na zbirki ImageNet, povsem zasenči druge tri. Izbrani prag na krivulji ROC se nahaja izjemno blizu teoretično optimalni točki (FPR, TPR) = (0, 1) v levem zgornjem kotu. Na histogramu izidov tudi vidimo, da mreža zelo dobro loči osebe z blago in resno okužbo. Cena, ki smo jo plačali za mrežo ResNet18* pa je seveda veliko število učnih parametrov in dolgotrajen čas učenja. Zanimivo je, da se je prazna mreža ResNet18 odrezala slabše od moje konvolucijske nevronske mreže, kar je verjetno posledica njene kompleksnosti in potrebnega časa učenja (ter optimizacije hiperparametrov), da dosežemo dobro učinkovitost.

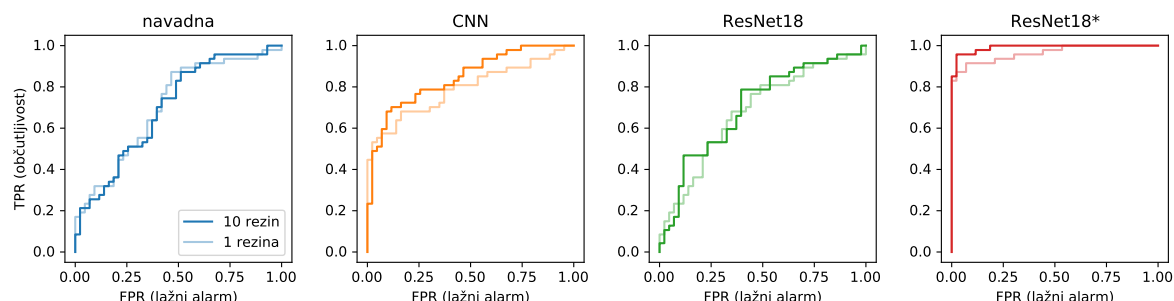


Slika 6: Primerjava uspešnosti vseh štirih najboljših nevronske mreže. Prikazane so krivulje ROC, natančnost-občutljivost in krivulja DET (prva vrstica), tri kombinirane mere uspešnosti v odvisnosti od izbrane vrednosti praga (druga vrstica) in histogram rezultatov z označenimi možnimi vrednostmi praga (tretja vrstica). Znaki (krožec, romb in križ) označujejo maksimume kombiniranih mer uspešnosti. Za izbrano vrednost praga (po vrednosti F_1) so prikazane tudi matrike zamenjav. V spodnjih vrsticah sta izpisana še število učljivih parametrov mreže in celotni čas učenja mreže.

navadna	70.0%	87.2%	48.8%	65.1%	74.5%
CNN	84.5%	68.1%	90.7%	77.1%	77.9%
ResNet18	70.0%	78.7%	60.5%	68.5%	73.3%
ResNet18*	99.1%	95.7%	97.7%	97.8%	96.8%
	AUC	občutljivost	specifičnost	natančnost	F1

Slika 7: Povzetek ključnih mer uspešnosti za vse štiri tipe nevronske mreže.

Zanimalo me je tudi, ali se morda mreže bolje obnesejo, če jih testiramo zgolj na centralnih rezinah. Na sliki 8 so prikazane krivulje ROC, na katerih vidimo, da se je zgolj navadna nevronska mreža nekoliko bolje obnesla pri testiranju na centralnih rezinah, medtem ko so ostale mreže bile uspešnejše, kadar so imele na voljo vseh 10 rezin.



Slika 8: Centralni vsi mreža

4.3 Samodiagnostika

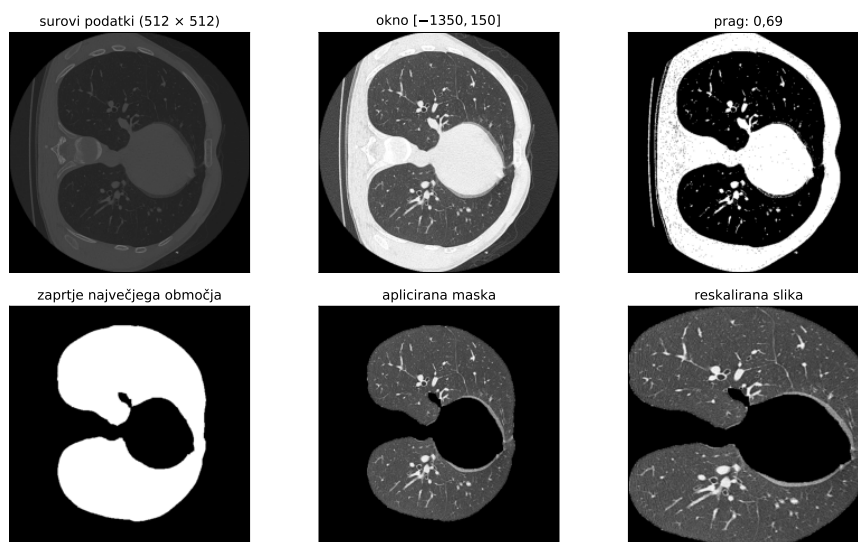
Marca 2016 sem tudi sam imel opravljen CT prsnega koša, zato sem se odločil preizkusiti delovanje nevronske mreže še na lastnih slikah. V tistem času COVID-19 seveda še ni obstajal, prav tako pa je moja diagnoza bila povsem drugačne narave kot pljučnice pri COVID-19. Svoje slike CT sem torej obravnaval kot primere *brez* okužbe, za katere bi smotrno pričakoval, da jih bo nevronska mreža razvrstila kot *blage* okužbe.

4.3.1 Predpriprava slik

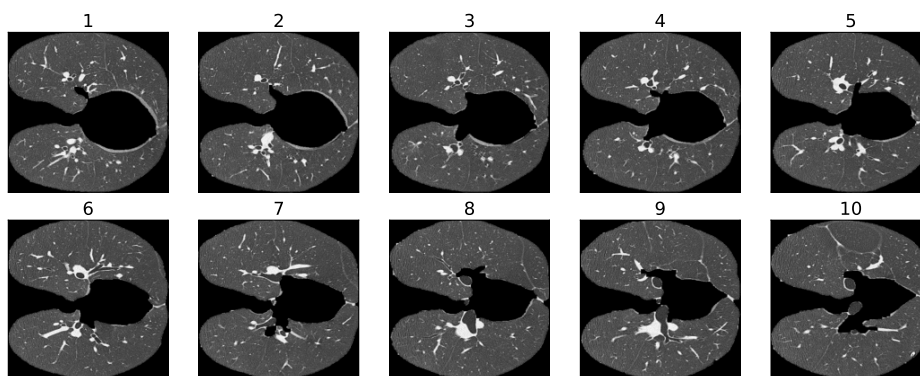
Slike so bile posnete 14. marca 2016 v UKC Maribor. Poleg slik so navedeni tudi obsevalni podatki, ki podajajo vrednost DLP (*dose length product*) 356,40 mGy cm, kar

lahko preračunamo v efektivno dozo 6,8 mSv oziroma 845 dni (2,3 leta) naravnega sevanja ozadja. Slike CT rekonstruirajo valj s premerom 29,5 cm in tvorijo rezine debeline 0,5 mm. Slike so bile shranjene v formatu DICOM, zato sem jih najprej pretvoril v format NIfTI s programom MRICroGL. Dobljene slike sem nato v Pythonu ustrezno pripravil na enak način, kot je opisano v navodilih naloge. Pri tem sem uporabil prilagojeno izvirno kodo avtorjev članka iz revije *Nature*¹. Surove podatke velikosti 512×512 pikslov sem normaliziral z ustreznim oknom za pljuča $[-1350, 150]$. Nato sem sliko binariziral z empirično izbranim pragom 0,69. Na dobljeni binarni sliki je algoritem poiskal povezana območja. Največjega izmed območij, ki ni ozadje, je algoritem nato še morfološko očistil in povezal (*morphological closure*) ter ga uporabil kot masko na normalizirani sliki. Dobljeno sliko sem nato še reskaliral na celotno velikost 512×512 . Postopek predpriprave slik je prikazan na sliki 9.

Izbral sem 10 rezin, ki so se po moji presoji nahajale na podobnih anatomskih višinah kot rezine iz podatkovnega seta MOSMED. Prikazane so na sliki 10.



Slika 9: Postopek predpriprave lastnih CT-slik.

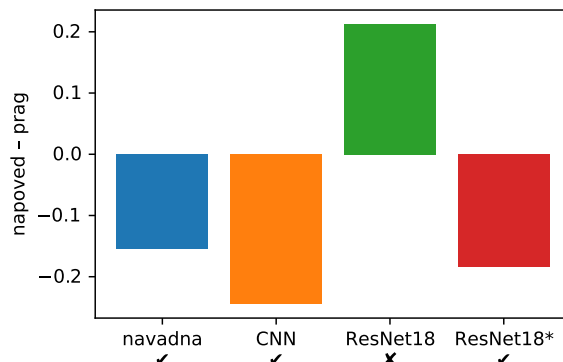


Slika 10: Izbranih 10 lastnih rezin.

Vseh 10 rezin sem nato na enak način kot testne podatke poslal v vsako izmed štirih najboljših nevronske mreže in izbral vrednost praga, kot sem jo določil pri primerjavi

¹https://github.com/howchihlee/COVID19_CT

mrež. Na sliki 11 so prikazani rezultati kot končna vrednost klasifikatorja minus izbrani prag. Pričakujemo torej *negativne* vrednosti. Opazimo, da je le mreža ResNet18 (brez predučenja) moje slike napačno razvrstila kot primer hude okužbe s COVID-19, ostale tri pa so pravilno napovedale "blago okužbo".



Slika 11: Rezultati klasifikacije lastnih slik CT.

5 Vizualizacija konvolucijskih mrež

Konvolucijske nevronske mreže se dobro izkažejo pri klasifikaciji slik, vendar si pogosto ne moremo intuitivno predstavljati, kako so abstraktni koncepti slik zapisani v utežeh mreže. Za boljšo predstavo lahko naredimo nekaj vizualizacij, s katerimi dobimo droben vpogled v delovanje konvolucijskih nevronskih mrež.

5.1 Vizualizacija konvolucijskih jeder

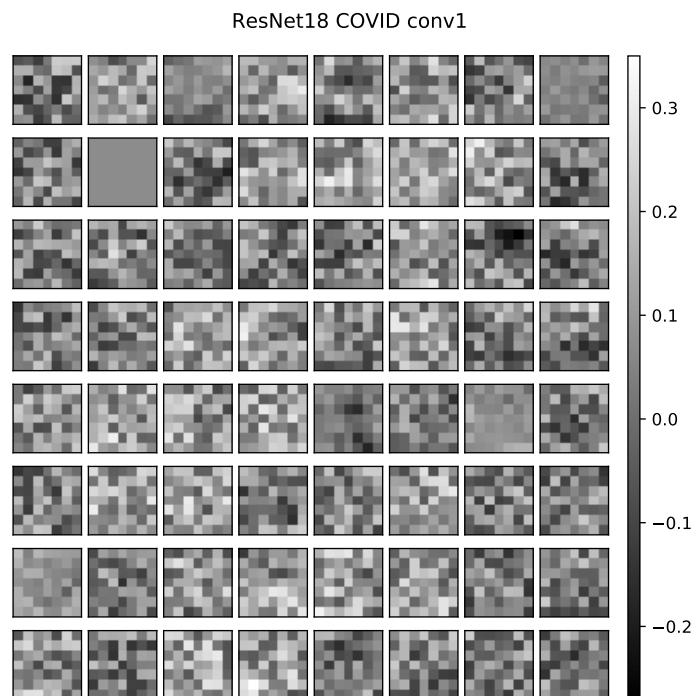
Na sliki 12 so prikazani filtri prve konvolucijske plasti (`conv1`) naučene mreže ResNet18*. Ko je mreža naučena na zbirki ImageNet, so v prvi plasti vidni filtri (konvolucijska jedra), ki iščejo prisotnost vzorcev, naprimer črt, pik, šahovnic. V tem primeru pa ne vidimo nobene jasne strukture, temveč bolj ali manj šum. To lahko razumemo v kontekstu vhodnih podatkov (in zelene klasifikacije). Pljučno tkivo blago obolelih in resno obolelih se ne razlikuje po prisotnosti nekih enostavnejših, značilnih linij ali obrisov, temveč po nejasnih, variabilnih spremembah v lokalni strukturi tkiva.

Na sliki 13 sem prikazal razliko med filtri druge konvolucijske plasti (`layer1[0].conv1`) med mrežo ResNet18*, naučeno na slikah CT, in naučeno na zbirki ImageNet. Vsekakor vidimo, da je mreža spremenila svoja konvolucijska jedra; v grobem bi lahko rekli, da se je vrednost večine filtrov ali v celoti povečala ali v celoti zmanjšala.

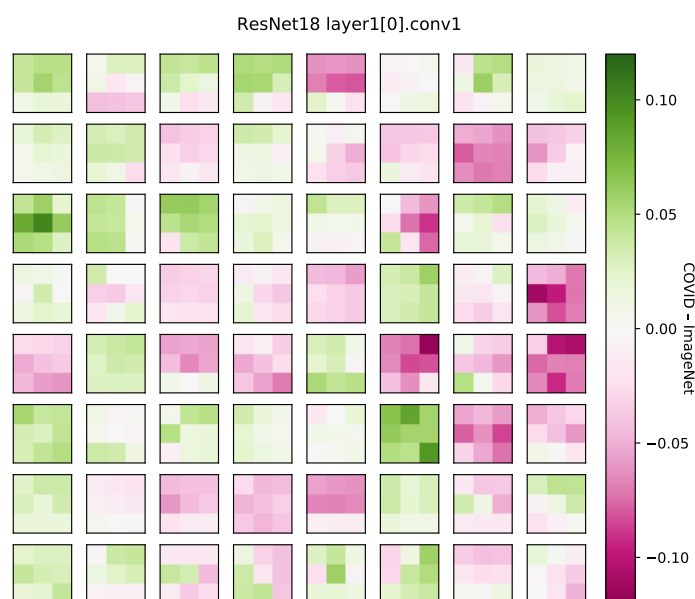
Za primerjavo sem vizualiziral tudi filtre lastne konvolucijske nevronske mreže, ki so prikazani na slikah 14 in 15. Ponovno opazimo zgolj šumnat filter, ki nam ne podaja veliko informacij.

5.2 Vizualizacija konvolucij

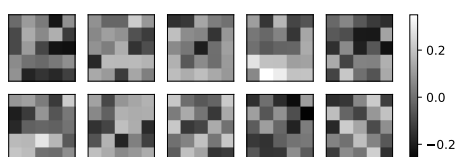
Ogledamo si lahko tudi, kako se vhodna slika spreminja, ko potuje skozi nevronske mreže in pri tem doživlja konvolucije z mnogimi jedri. Na slikah 16 in 17 je prikazano, kako se spreminjata slika z blago in slika s hudo okužbo. Ker je število plasti in konvolucijskih



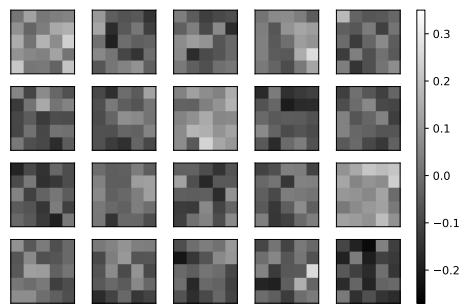
Slika 12: Filtri prve konvolucijske plasti mreže ResNet18*.



Slika 13: Razlika med filtri mreže ResNet18*, naučene na slikah CT in slikah ImageNet.

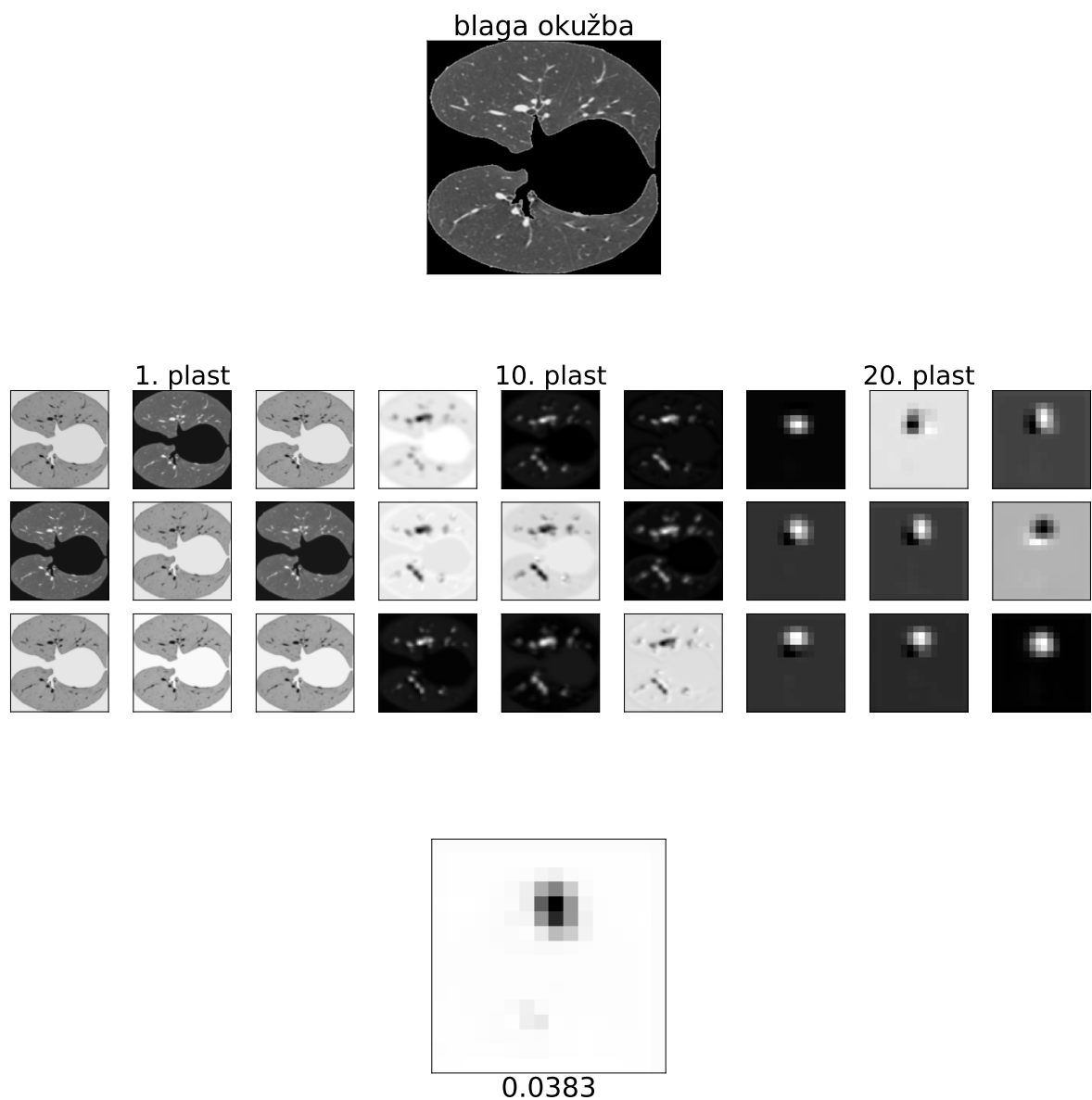


Slika 14: Filtri prve konvolucijske plasti mreže cnn2.

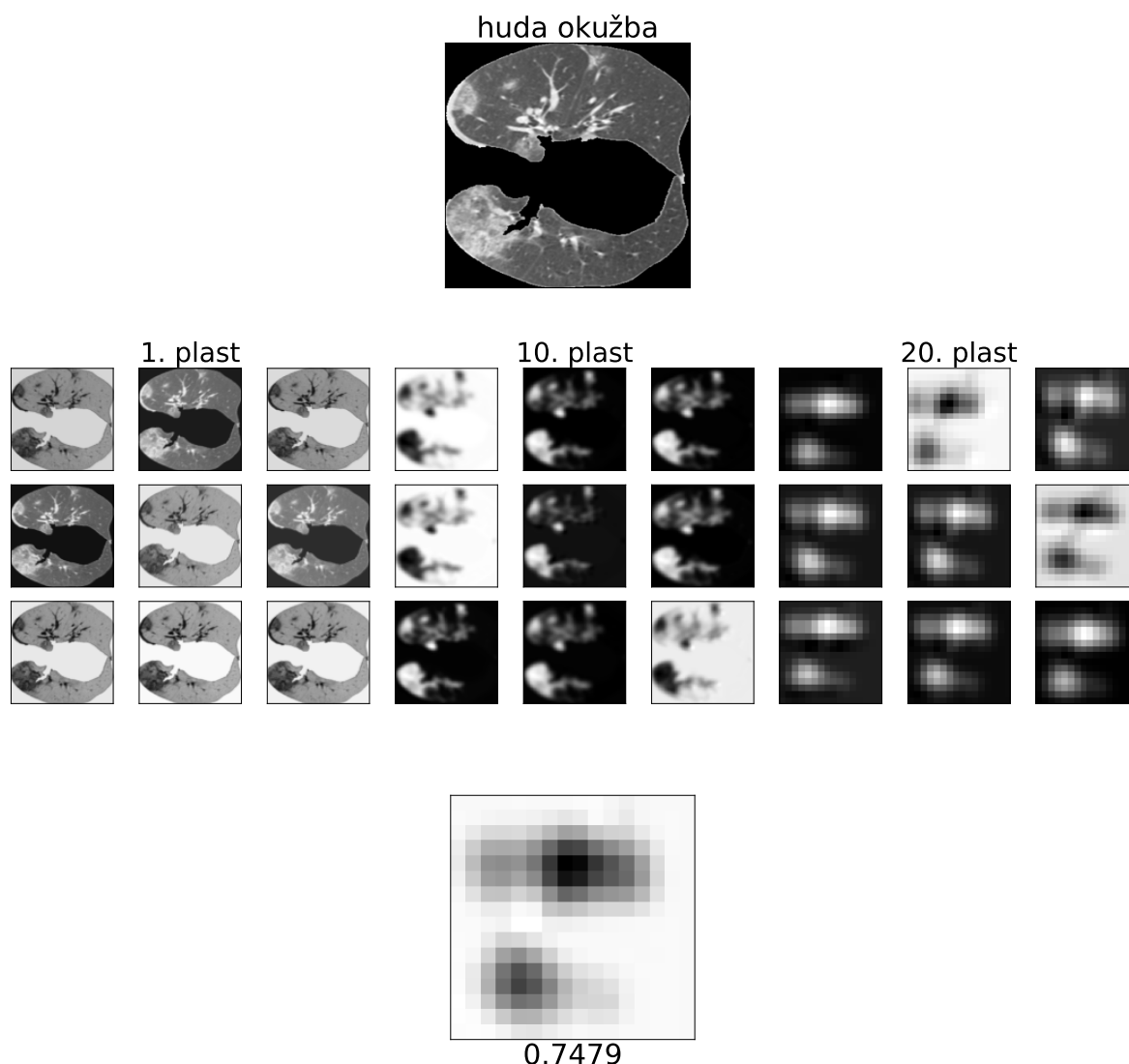


Slika 15: Filtri druge konvolucijske plasti mreže cnn2.

filtrorv ogromno, so prikazane zgolj tri plasti (prva, deseta in dvajseta), znotraj vsake pa le prvih rezultat konvolucije s prvimi devetimi filtri.



Slika 16: Spreminjanje vhodne slike z blago okužbo med potovanjem po nevronske mreži ResNet18*.

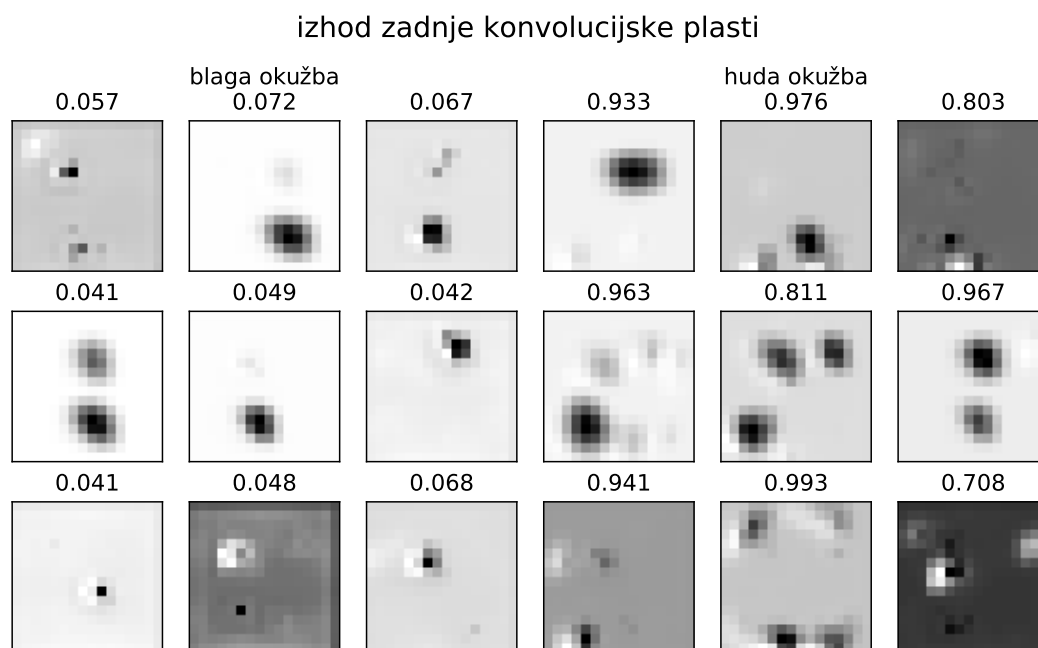


Slika 17: Spreminjanje vhodne slike z resno okužbo med potovanjem po nevronske mreži ResNet18*.

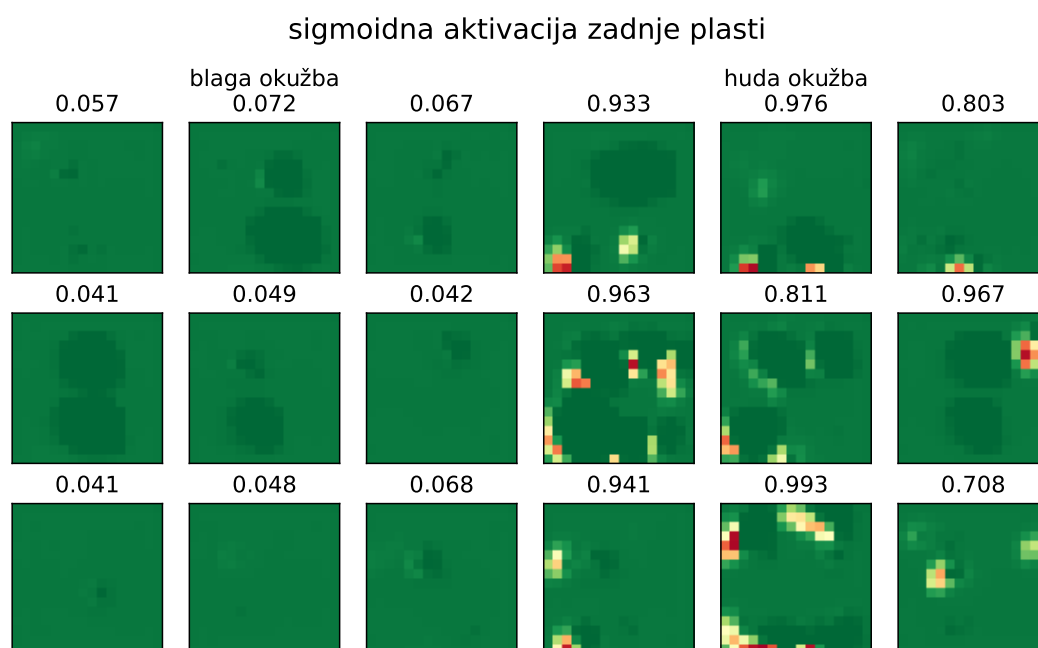
Zanimalo me je, kako se bo v splošnem izhodna slika zadnje konvolucijske plasti pri blagih okužbah razlikovala od resnih okužb. Na sliki 18 je prikazana primerjava, vendar slike niso izrisane v isti sivinski skali. Takšna primerjava nam ne pove veliko: v ozir moramo seveda vzeti, da po zadnji konvolucijski plasti sledi enostaven zbiralnik MaxPool in nato še sigmoidna aktivacijska funkcija! Na sliki 19 so tako prikazane iste slike, tokrat aktivirane s sigmoidno funkcijo in prikazane na isti barvni skali. Jasno vidimo, da se pri hudih okužbah pojavljajo pikseli z visokimi vrednostmi – najvišja vrednost piksla seveda sovpada s končnim rezultatom nevronske mreže.

5.3 Vizualizacija izpostavljenosti

Dober vpogled v delovanje konvolucijskih nevronske mrež dobimo z uporabo knjižnice Captum, ki omogoča izračun različnih map izpostavljenosti. Prikazal bom tri različne algoritme. Prvi je izpostavljenost (*saliency*), pri katerem učinkovito izračunamo odvod končnega rezultata glede na vhodno sliko (za vsak vhodni piksel). S tem ugotovimo, ko-



Slika 18: Primerjava izhoda zadnje konvolucijske plasti pri blagih okužbah (levo) in hudih okužbah (desno). Slike med seboj niso prikazane v isti sivinski skali!

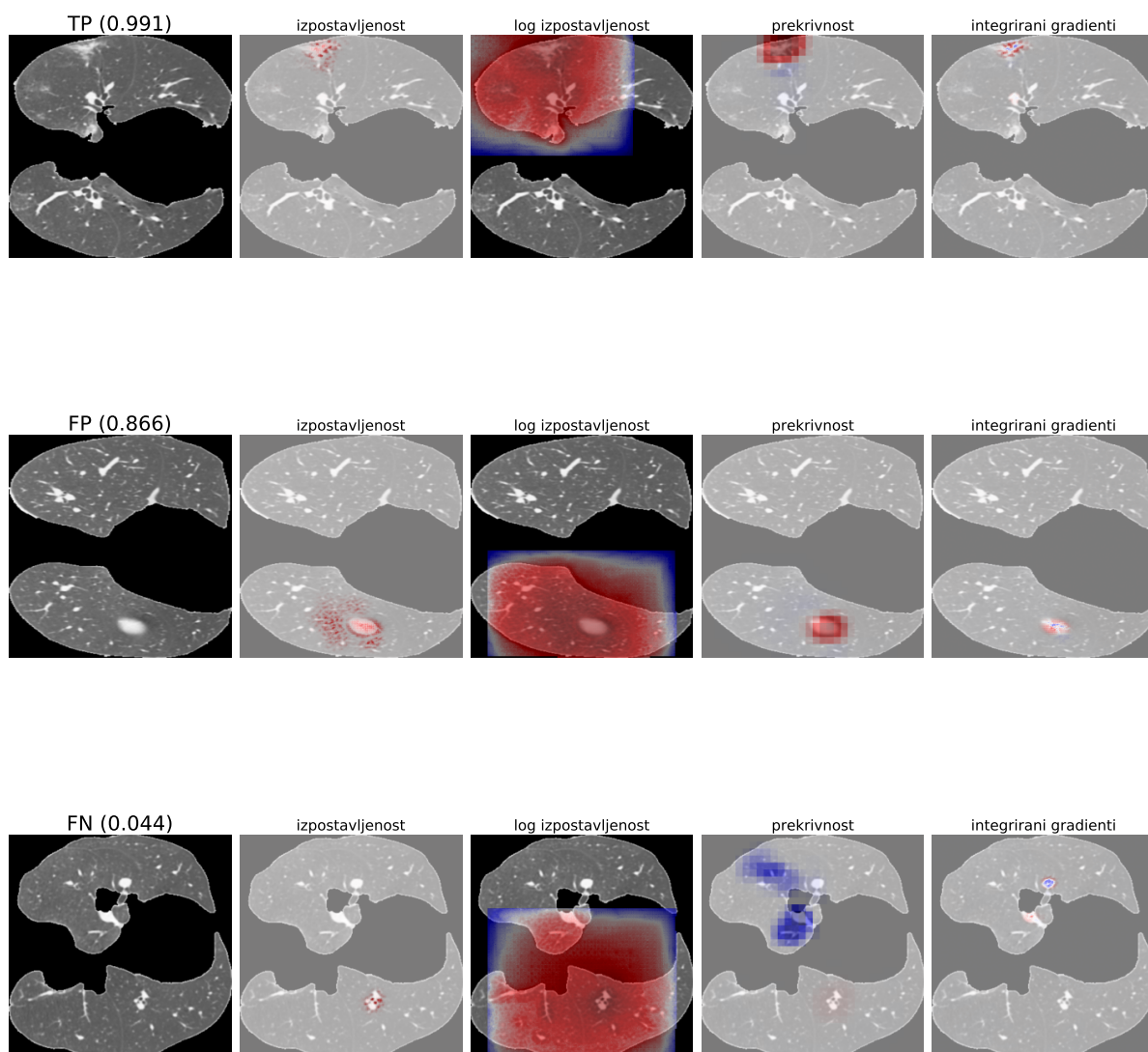


Slika 19: Izhod zadnje konvolucijske plasti, aktiviran s sigmoidno funkcijo. Slike so prikazane na isti barvni skali od 0 (zeleno) do 1 (rdeče).

likšen je doprinos posameznega piksla vhodne slike h končnemu rezultatu oziroma koliko (majhna) sprememba piksla vpliva na končni rezultat. Drugi algoritem je prekrivnost (*occlusion*), ki deluje tako, da po vhodni sliki pomika črn kvadrat, s katerim zakrije del slike, pri tem pa vsakič izračuna rezultat klasifikatorja na takšni delno prekriti sliki. S tem ugotovimo, kateri deli slike pomembno vplivajo na končni rezultat. Tretji algoritem

so integrirani gradienti (*integrated gradients*), pri katerih podobno kot pri izpostavljenosti izračunamo odvod vsakega piksla, le da tokrat ne izračunamo zgolj odvoda pri trenutni vrednosti piksla, temveč piksel ugasnemo in po korakih postopoma prižigamo do končne vrednosti. Na vsakem koraku izračunamo odvod, nato pa vse odvode seštejemo (kar bi v zvezni limiti prineslo integriranje). S tem dobimo boljši vpogled v pomembnost piksla, saj obravnavamo celotno območje vrednosti, ki jih piksel lahko zavzame in ne le okolice njegove dejanske vrednosti.

Na sliki 20 so prikazane mape izpostavljenosti za različne izide nevronske mreže ResNet18*. Opazimo, da se vse tri metode običajno ujemajo v položaju najpomembnejšega dela slike. Pri lažno pozitivnem izidu je nevronska mrežo zmotila nejasna bela lisa, ki je morda zgolj artefakt. Omeniti je treba, da so prikazani rezultati zgolj za eno rezino, medtem ko je končni rezultat mreže kombinacija (povprečje) 10 rezin. Lažno pozitiven rezultat na zgolj eni rezini lahko tako kompenzirajo pravilno negativni rezultati drugih rezin.



Slika 20: Mape izpostavljenosti za nevronska mrežo ResNet18*.

6 Zaključek

V nalogi smo obravnavali različne arhitekture nevronske mreže, ki smo jih uporabili kot klasifikatorje za ločevanje resnosti okužbe COVID-19 na podlagi predobdelanih CT-slik prsnega koša. Primerjali smo različne strukture in pokazali znatno prednost predučene konvolucijske nevronske mreže ResNet18 pred drugimi arhitekturami. Ukvarjali smo se tudi s problemom merjenja uspešnosti binarnih klasifikatorjev in predstavili različne mere uspešnosti. Izučene nevronske mreže smo preverili tudi na lastnih CT-slikah. V drugem delu smo poskušali s tremi različnimi pristopi vizualizirati delovanje nevronske mreže. Največ informacij so nam dale mape izpostavljenosti, ki kažejo, katera območja na sliki pomembno prispevajo h končnemu rezultatu.

Za konkretni primer klasifikacije (ali celo diagnostike) okužbe COVID-19 bi sicer bila uporabnejša nevronska mreža, ki bi kot vhodno sliko prejela zgolj rentgensko sliko prsnega koša, saj je takšna preiskava hitrejša, predvsem pa mnogo varnejša za bolnika.