

# Uspostavljanje dijagnoze primjenom umjetnih neuronskih mreža na snimkama magnetne rezonance mozga

---

Lukša, Ana

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:165084>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-12-24**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



**SVEUČILIŠTE JURJA DOBRILE U PULI  
FAKULTET INFORMATIKE**

**DIPLOMSKI RAD**

**Uspostavljanje dijagnoze primjenom  
umjetnih neuronskih mreža na  
snimkama magnetne rezonance  
mozga**

Ana Lukša

Pula, rujan 2022.  
SVEUČILIŠTE JURJA DOBRILE U PULI  
FAKULTET INFORMATIKE

DIPLOMSKI RAD  
**Uspostavljanje dijagnoze primjenom  
umjetnih neuronskih mreža na  
snimkama magnetne rezonance  
mozga**  
Ana Lukša

JMBAG: 1003120682, izvanredni student

Studijski smjer: Informatika

Kolegij: Umjetna inteligencija

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informacijski sustavi i informatologija

Mentor: izv. prof. dr. sc. Darko Etinger



## IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisana Ana Lukša, kandidat za magistru informatike ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

*Ana Lukša*

U Puli, rujan. 2022. godine



## IZJAVA o korištenju autorskog djela

Ja, Ana Lukša dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom „ Uspostavljanje dijagnoze primjenom umjetnih neuronskih mreža na snimkama magnetne rezonance mozga “ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama. Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, rujan. 2022.

Potpis

Ana Lukša

# Zahvala

Zahvaljujem svojem mentoru, izv. prof. dr. sc. Darku Etingeru na korisnim savjetima i smjernicama tijekom izrade ovog rada. Zahvaljujem svojoj obitelji na pruženoj podršci i strpljenju tijekom cijelog studija.

# Sadržaj

|   |    |
|---|----|
| 1. Uvod .....   | 1  |
| 2. Radiologija .....  | 3  |
| 2.1. Magnetska rezonanca.....   | 3  |
| 2.1.1. MRI tehnika.....   | 6  |
| 2.2. Tumori mozga.....  | 7  |
| 2.2.1. Glioblastomi .....   | 7  |
| 2.2.2. Meningeomi .....   | 8  |
| 2.2.3. Tumori hipofize .....  | 9  |
| 3. Umjetna inteligencija .....  | 10 |
| 3.1. Strojno učenje.....  | 11 |
| 3.2. Duboko učenje.....   | 12 |
| 3.3. Umjetne neuronske mreže .....  | 14 |
| 3.3.1. Vrste umjetne neuronskih mreža .....   | 15 |
| 3.4. Primjeri kliničkih područja primjene umjetne inteligencije u radiologiji ..... | 17 |
| 3.4.1. Klasifikacija .....  | 20 |
| 3.4.2. Segmentacija .....   | 20 |
| 3.4.3. Detekcija.....   | 21 |
| 4. Konvolucijska neuronska mreža .....  | 23 |
| 4.1. Konvolucijski sloj .....   | 24 |
| 4.1.1. Nadopunjavanje.....  | 27 |
| 4.1.2. Aktivacijska funkcija .....  | 29 |
| 4.2. Sloj sažimanja.....  | 30 |
| 4.2.1. Sažimanje maksimalnom vrijednošću .....                                      | 30 |
| 4.2.2. Sažimanje srednjom vrijednošću .....   | 31 |
| 4.3. Potpuno povezani sloj.....   | 32 |
| 4.4. Treniranje mreže .....   | 33 |
| 4.5. Funkcija gubitka.....  | 34 |
| 4.6. Prenaučenost i podučenost.....   | 35 |

|   |    |
|---|----|
| 4.7. Regularizacija .....                 | 36 |
| 4.7.1. Ispadanje neurona.....             | 36 |
| 4.7.2. Rano zaustavljanje .....           | 37 |
| 4.7.3. Augumentacija podataka .....       | 37 |
| 4.7.4. Normalizacija grupe.....           | 38 |
| 4.8. Optimizacijski algoritmi .....       | 39 |
| 4.8.1. Gradijentni spust.....             | 39 |
| 4.8.2. Stohastički gradijentni spust..... | 41 |
| 4.8.3. Opadanje stope učenja.....         | 42 |
| 5. Grad-CAM.....                          | 43 |
| 6. Transferno učenje .....                | 47 |
| 6.1. Arhitekture mreža .....              | 49 |
| 6.1.1. EfficientNet.....                  | 49 |
| 6.1.2. Resnet .....                       | 51 |
| 6.1.3 Inception .....                     | 53 |
| 6.1.4. Xception .....                     | 55 |
| 7. Metode rada.....                       | 58 |
| 7.1. Skup podataka i predobrada .....     | 58 |
| 7.2. Izrada i treniranje modela .....     | 61 |
| 7.3. Evaluacija i testiranje modela.....  | 65 |
| 8. Zaključak .....                        | 69 |
| Literatura .....                          | 70 |
| Popis slika .....                         | 74 |
| Sažetak.....                              | 77 |
| Abstract.....                             | 78 |



# 1. Uvod

Tijekom posljednjih desetljeća svjedočili smo važnosti medicinskog snimanja, za rano otkrivanje, dijagnosticiranje i liječenje bolesti. Radiološko snimanje ima veliki značaj u njezi bolesnika, a dijagnostičke i terapijske indikacije za radiološko snimanje brzo se šire. Brza ekspanzija posljedica je potrebe za bržim, točnim, isplativijim i manje invazivnim liječenjem. Tehnološki napredak u opremi za radiološke slike također je potaknuo korištenje snimanja. Takva tehnološka dostignuća uključuju mogućnost dobivanja slika sve veće rezolucije, omogućujući vizualizaciju manjih anatomskih struktura i abnormalnosti. Veća razlučivost dolazi po cijenu sve većeg prosječnog broja slika po pacijentu. Te slike različitih modaliteta trebaju interpretirati radiolozi, obično u kratkom vremenu, a kako se broj i složenost slika povećava, povećava se i radno opterećenje radiologa. [1]

Iako se, u usporedbi s napretkom u medicinskim slikovnim tehnologijama, kasni s napretkom u računskoj analizi medicinske slike, nedavno se poboljšava uz pomoć tehnika strojnog učenja. Napredak u strojnom učenju, posebno u načinu dubokog učenja, omogućio je veliki skok kako bi pomogao identificirati, klasificirati i kvantificirati obrasce u medicinskim slikama. Konkretno, iskorištavanjem hijerarhijskih prikaza značajki naučenih isključivo iz podataka, umjesto ručno izrađenih značajki koje su uglavnom dizajnirane na temelju znanja specifičnog za domenu, duboko učenje brzo se pokazuje kao najsuvremeniji temelj, postižući poboljšane performanse u različitim medicinskim primjenama. [2]

Najutvrđeniji algoritam među različitim modelima dubokog učenja je konvolucijska neuronska mreža (CNN), vrsta umjetnih neuronskih mreža koja je dominantna metoda u zadacima računalnog vida otkako su zapanjujući rezultati podijeljeni na natjecanju za prepoznavanje objekata poznatom kao ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012. Medicinska istraživanja nisu iznimka, jer je CNN postigao vrhunske rezultate u različitim područjima, primjerice probir dijabetičke retinopatije, klasifikaciju kožnih lezija i otkrivanje metastaza u limfnim

čvorovima. Među istraživačima radiologije došlo je do porasta interesa za potencijal CNN-a, a već su objavljene mnoge studije u područjima kao što su otkrivanje lezija, kompjuterska dijagnoza i prognoza, klasifikacija, segmentacija slike, rekonstrukcija slike i obrada prirodnog jezika. [3]

Strojno učenje pruža učinkovit način za automatizaciju analize i dijagnoze medicinskih slika. To potencijalno može smanjiti opterećenje radiologa uz uštedu rada i točne dijagnostičke rezultate. Osim toga, zbog velikih varijacija u patologiji i potencijalnog umora ljudskih stručnjaka, istraživači i liječnici nedavno su počeli imati koristi od računalno potpomognutih intervencija. [1, 2]

Razlikovanje tumora mozga vrlo je izazovan zadatak u velikoj prostornoj i strukturnoj varijabilnosti okolne regije tumora mozga. Stoga je svrha ovog rada razviti model koji se temelji na umjetnoj inteligenciji (AI), odnosno konvolucijskoj neuronskoj mreži (CNN), da prepozna tri najčešća tipa tumora mozga, gliom, meningeom i tumor hipofize pomoću slika magnetske rezonancije (MR) te ispitati četiri arhitekture neuronskih mreža, EfficientNet, ResNet50v2, Inception te Xception. Prvi dio rada opisuje princip rada magnetske rezonance te prikaz najčešćih tumora mozga, dok su u drugom obrađeni pojmovi umjetne inteligencije, strojnog i dubokog učenja, umjetne neuronske mreže te njihove primjene u radiologiji. Arhitektura i način rada konvolucijske neuronske mreže objašnjeni su u trećem dijelu. Nakon toga slijedi opis metode razvoja, treniranja i testiranja mreže na skupu podataka sa stranice Kaggle.com te dobiveni rezultati.

## 2. Radiologija

Radiologija je grana medicinske znanosti koja koristi tehnologiju snimanja i zračenja za postavljanje dijagnoze i liječenje bolesti. Imala je velike koristi od napretka fizike, elektroničkog inženjerstva i računalne znanosti. Na temelju različitih razloga za otkrivanje i slikovnu obradu, u proteklim desetljećima razvijeni su različiti modaliteti u području dijagnostičke radiologije. Danas, glavni modaliteti koji se široko koriste u bolnicama i zdravstvenim ustanovama uključuju radiografiju, fluoroskopiju, kompjuteriziranu tomografiju (CT), ultrazvuk, magnetsku rezonancu (MRI) i pozitronsku emisijsku tomografiju (PET). [1]



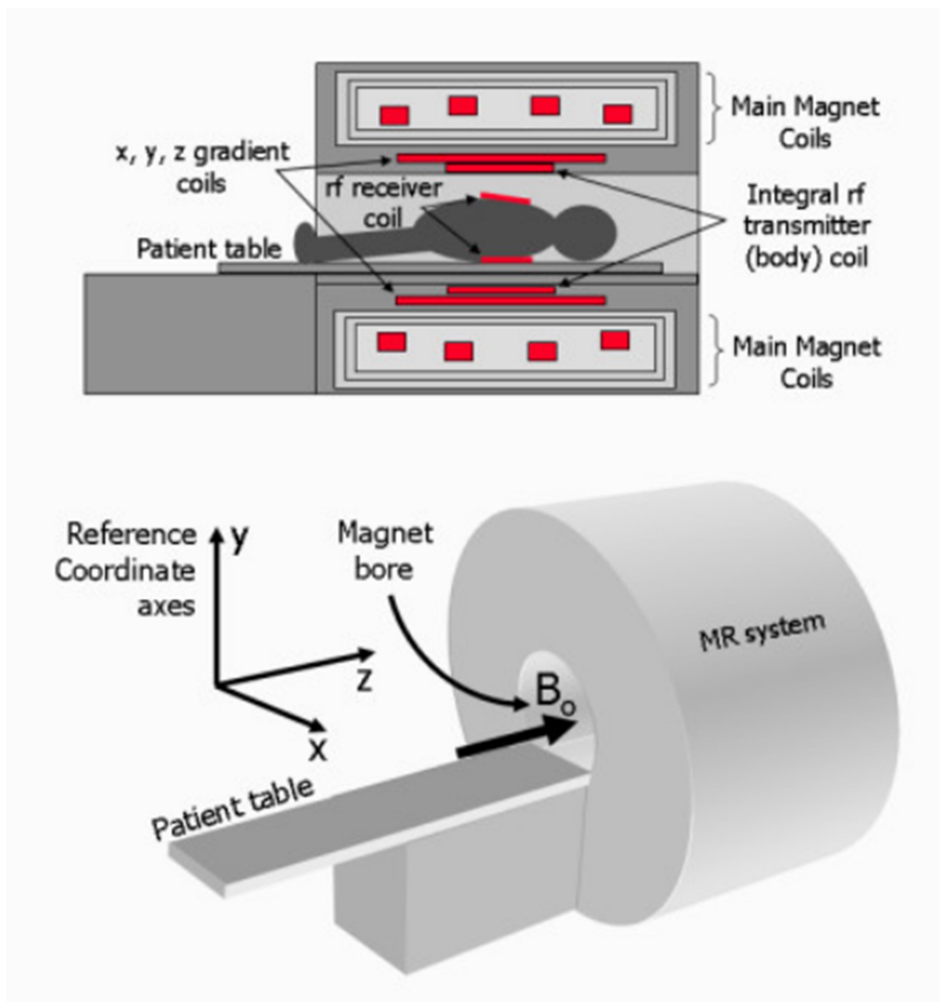
Slika 1: primjeri radioloških snimki različitih modaliteta

### 2.1. Magnetska rezonanca

MRI predstavlja oblik računalne slike s  $n \times m$  numeričkih vrijednosti, odnosno piksela predstavljenih na zaslonu kao skala sive boje pomoću procesa digitalno-analogne konverzije. Intenzitet prikazanog piksela proporcionalan je njegovoj numeričkoj vrijednosti i odražava kumulativnu snagu radiofrekvencijskog (RF) signala primljenog od odgovarajućeg elementa volumena (voksela) unutar presjeka tkiva koje se ispituje. Ovaj RF signal ovisi o čimbenicima: snaga magnetskog polja, veličina i položaj RF detektora (antena, zavojnica, površinska zavojnica), parametri relaksacije

(T1 i T2) tkiva, količina mobilnih protona u tkivu (njegova gustoća protona), parametri MRI snimanja. [4]

U MRI snimanju, pacijent se stavlja u vanjsko magnetsko polje MRI magneta. Magnetski dipolni moment vodikovih jezgri u tijelu, koje su inače nasumično orijentirane u slabom magnetskom polju Zemlje, pod utjecajem je ovog mnogo jačeg vanjskog polja i podvrgnuti su neto poravnanju paralelnom s poljem. Pri ovoj jakosti polja, samo oko 2 do 4 u svaka 2 milijuna protona pacijenta doprinosi MRI signalu; svi ostali su poništeni protonima usmjerenim u suprotnom smjeru. [4]



Slika 2: uređaj za magnetsku rezonancu

Pojedinačno, protonski magnetski momenti precesiraju oko statičkog polja, ali se mogu zajedno procijeniti tako da je neto magnetizacija u početku usmjerena duž statičkog polja. Kada su podvrgnute kratkim razdobljima RF energije, jezgre vodika u tkivima pacijenta apsorbiraju RF energiju i mijenjaju svoju orijentaciju. U ovom novom poravnanju može se pokazati da neto magnetizacija precesira oko statičkog polja. Apsorbirana RF energija i naknadna precesija i preusmjeravanje jezgri omogućuju im da funkcioniraju kao minijturni RF odašiljači dok se vraćaju na svoje prethodno paralelno poravnanje. Njihov RF signal se mjeri (prima) pomoću vanjske RF antene (zavojnice). [4]

Ponavljajuće sustavne varijacije u jakosti drugog magnetskog polja (generiranog gradijentnim zavojnicama unutar magneta) tijekom vremena kada su jezgre stimulirane vanjskim RF signalom omogućuje selektivno pobuđivanje pojedinačnih područja tijela. Snimanje i naknadna obrada ovih prostorno lokaliziranih podataka rezultira kliničkom MRI snimkom. [4]

Povratak magnetizacije na ravnotežno poravnanje i ravnotežnu vrijednost slijedi eksponencijalnu funkciju vremena. Budući da različita tkiva stvaraju različita fizička i elektrokemijska okruženja, njihovo se opuštanje događa različitim brzinama, a njihov eksponencijalni oporavak opisuje se njihovim vremenima opuštanja, T1 (engl. inversion time). Dakle, različita tkiva imaju različito longitudinalno vrijeme opuštanja, T1. Kada se magnetizacija usmjeri u poprečnu ravninu, dobiva se FID (inducirani napon iz magnetizacije bez RF pulsa). Pažljivo ispitivanje FID-a otkriva da se njegova amplituda eksponencijalno smanjuje s vremenom. Stoga se može pokazati da transverzalna magnetizacija također opušta, a parametar, T2, koristi se za opisivanje brzine ovog eksponencijalnog gubitka signala u transverzalnoj ravnini. Različita tkiva imaju različite T2, baš kao što se razlikuju po T1 i gustoći protona. Stoga, snaga MR signala u bilo kojem trenutku tijekom njegove indukcije ovisi o tri parametra - gustoći protona, T1 i T2; ovi parametri su odgovorni za kontrast u MRI slikama. [4]

Matematički proces nazvan Fourierova transformacija koristi se za pretvaranje informacija o frekvenciji sadržanih u signalu sa svake lokacije u snimljenoj ravnini u odgovarajuće razine intenziteta, koje se zatim prikazuju kao nijanse sive u matičnom rasporedu, na primjer, 256 X 256 piksela. [4]

### **2.1.1. MRI tehnika**

MR snimanja mozga izvode se u aksijalnoj, koronarnoj i sagitalnoj osi u različitim debljinama od 5 do 10 mm. MRI snimanje općenito se izvodi u aksijalnim presjecima, paralelno s kantomeatalnom linijom. Ovi snimci se dijele na rezove stražnje jame u pomacima od 5 mm i supratentorijalne rezove u pomacima od 8 mm. MRI je napravio revoluciju u ovom području svojom sposobnošću dobivanja sagitalnih i koronarnih slika bez posebnog pozicioniranja pacijenta. Stoga je pacijentima udobnije, a slike su manje osjetljive na artefakte pokreta koji mogu nastati zbog posebnog pozicioniranja.

T1 slike imaju kraće vrijeme ponavljanja (engl. time to repeat, TR) manje od 1000 msec i vrijeme odjeka (engl. time to echo, TE) manje od 30 msec. T2-ponderirane slike imaju dugi TR od više od 1500 msec. Prvi odjek T2 slika s kraćim TE naziva se protonska gustoća ili uravnotežene T1 i T2 slike. Drugi odjek T2 slika s dužim TR od preko 60 msec predstavlja prave T2 slike. Istraživanja su pokazala da većina patoloških procesa uzrokuje povećanje sadržaja vode u mozgu. Stoga su T2 slike vrlo osjetljive u otkrivanju patologije mozga. Rutinski se izvode T1 i T2 aksijalne i T2 koronarne studije. Slike T2 sadrže prvi odjek s kratkim TR i drugi odjek dugog TR. [4]

Likvor ima dugi T1, koji pokazuje nizak intenzitet signala u T1 slikama, ali postaje nešto viši u intenzitetu u prvim eho T2 slikama. U stvarnim T2 (dugi TE i vrlo dugi TR) slikama, likvor ima vrlo visok intenzitet signala. Bazalni gangliji, nazubljene jezgre malog mozga i crvene jezgre srednjeg mozga imaju niske intenzitete signala u T2 slikama zbog svog sadržaja minerala. Siva tvar ima nešto veći intenzitet signala u odnosu na bijelu tvar u gotovo svim spin-echo slikama. Vrlo visok intenzitet signala masti na T1 slikama može se cijeliti unutar diploičkog prostora i ocrtan je niskim intenzitetom

signala kortikalne kosti. Vaskularne strukture pokazuju prazan signal na svim pravilnim T1, T2 i slikama gustoće protona zbog pokretnih protona cirkulirajuće krvi. Likvor može imati isti učinak, pokazujući signal praznina u cerebralnom akvaduktu J9. Izgled žila može se promijeniti ako se primijeni posebna tehnika magnetske rezonancije (MRA). Do sada su spojevi gadolinija jedini kontrastni medij odobren od strane Uprave za hranu i lijekove koji se može koristiti intravenozno. [4]

## **2.2. Tumori mozga**

Intrakranijalni tumori mogu zahvatiti mozak i druge strukture kao što su moždani živci i ovojnice. Neki tumori su benigni, ali zbog lubanjskog svoda koji ne ostavlja mjesta za širenje, mogu uzorkovati gubitak nekih kritičnih funkcija kao što su razmišljanje, govor i pokret, a ponekad i smrt. Rano otkrivanje moždanih bolesti može pomoći da se na vrijeme dobije najbolji tretman. Jedna od konvencionalnih metoda za dijagnosticiranje ovih poremećaja je tehnika magnetske rezonance.

### **2.2.1. Glioblastomi**

Glioblastomi su najčešći primarni tumori mozga kod odraslih, agresivni su, relativno otporni na terapiju i imaju lošu prognozu. Tipično se pojavljuju kao heterogene mase sa središtem u bijeloj tvari s nepravilnim perifernim povećanjem, središnjom nekrozom i okružene vazogenim edemom, oteklinom ili upalom. Iako glioblastomi mogu nastati bilo gdje u mozgu, oni imaju sklonost prema supkortikalnoj bijeloj tvari i dubokoj sivoj tvari moždanih hemisfera, osobito temporalnog režnja. Obično su veliki tumori u trenutku dijagnoze.

Na T1 je hipo do izointenzivna masa unutar bijele tvari, a središnji je heterogeni signal (nekroza, intratumorsko krvarenje). Kontrastom je poboljšanje varijabilno, ali je gotovo uvijek prisutno. Mikrovaskularna proliferacija rezultira obiljem novih žila s loše formiranom krvno-moždanom barijerom koja dopušta curenje gadolinija u susjedni izvanstanični intersticij što dovodi do uočenog poboljšanja na MRI-u.

Na T2/FLAIR je hiperintenzivan, okružen vazogenim edemom, a povremeno se vide šupljine protoka.

Glioblastome se općenito pažljivo prati s magnetskom rezonancom. Iako će vrijeme i učestalost varirati između ustanova i kirurga/onkologa zaduženih za liječenje pacijenta, općenito se snimanje obavlja unutar 24-48 sati od operacije kako bi se procijenila rezidualna bolest (prije nego se razvije postoperativno poboljšanje), a nakon toga svakih 8 do 12 tjedana. [5, 6]

## **2.2.2. Meningeomi**

Meningeomi predstavljaju najčešći tumor moždanih ovojnica, a većina ih je benigna. Oni su neoplazma koja potječe iz meningocita ili stanica arahnoidne kapice moždanih ovojnica i nalaze se svugdje gdje se nalaze moždane ovojnice, a na nekim mjestima gdje se pretpostavlja da se nalaze samo stanice mirovanja.

Snimanjem magnetskom rezonancom, tipično se čine izo- do hipointenzivnim na T1-ponderiranom prikazu (T1WI) i izo- do hiperintenzivnim na T2-ponderiranom prikazu (T2WI). Kod meningeoma, kontrast na T1-ponderiranim slikama se povećava intravenskom primjenom kontrastnog sredstva na bazi gadolinija jer se na taj način najpreciznije ocrta tumor, prisutnost unutarkoštanog i transkoštanog proširenja i odnos s mozgom. Strukturna magnetska rezonanca s kontrastom (MR) rutinski se obavlja u pacijenata s meningeomom za definiranje opsega tumora, planiranje liječenja i praćenje nakon liječenja, posebno za dijagnozu recidiva tumora. Dodatno, kompjutorizirana tomografija (CT) omogućuje, osim identificiranja kalcifikacija za diferencijalnu dijagnozu, dijagnozu zahvaćenosti kostiju susjedne kosti lubanje, što je od posebnog značaja za razgraničenje meningeoma i donošenje odluka o liječenju te kada pacijenti ne mogu obaviti pretragu magnetskom rezonancom, a posebno kada je meningeom potpuno okoštao/kalcificiran. [7, 8]



### **2.2.3. Tumori hipofize**

Tumori hipofize obično se otkriju kada osoba ode liječniku zbog simptoma koje ima. Ali ponekad ti tumori ne uzrokuju simptome, a pronalaze se tijekom medicinskih pretraga za druge zdravstvene probleme.

MR snimanje vrlo je korisno u pregledu mozga i smatra se najboljim načinom za pronalaženje tumora hipofize svih vrsta. MR slike obično su detaljnije od onih iz CT snimanja. Mogu prikazati makroadenome hipofize, kao i većinu mikroadenoma. Ponekad će magnetska rezonanca pokazati malu promjenu u hipofizi koja nema nikakve veze s pacijentovim simptomima. Između 5% i 25% zdravih ljudi ima neku manju abnormalnost hipofize koja se vidi na MRI snimci. [9]

Ručna dijagnoza abnormalnosti mozga je dugotrajna i teško je uočiti sitne promjene na MR slikama, osobito u ranim fazama abnormalnosti. Stoga su se modeli dubokog učenja naširoko koristili za analizu medicinske slike tijekom posljednjih nekoliko godina.

### 3. Umjetna inteligencija

Umjetna inteligencija (engl. artificial intelligence, AI) je polje koje kombinira informatiku i velike skupove podataka kako bi se omogućilo rješavanje problema. Također obuhvaća podpodručja strojnog učenja i dubokog učenja, koja se često spominju u vezi s umjetnom inteligencijom. Ove discipline se sastoje od AI algoritama koji nastoje stvoriti ekspertne sustave koji predviđaju ili klasificiraju na temelju ulaznih podataka. [10]

Oponaša ljudsku inteligenciju i koristi se za predviđanje, automatizaciju i optimizaciju zadataka, kao što su govor i prepoznavanje objekata, donošenje odluka i prevođenje. Postoje tri glavne kategorije AI: umjetna uska inteligencija (engl. artificial narrow intelligence, ANI), opća umjetna inteligencija (engl. artificial general intelligence, AGI) i umjetna super inteligencija (engl. artificial super intelligence, ASI). ANI se smatra "slabom", dok su druga dva tipa klasificirana kao "jaka" umjetna inteligencija. Uska umjetna inteligencija je osposobljena i fokusirana na obavljanje određenih zadataka, poput pobjede u partiji šaha ili identificiranja određene osobe u nizu fotografija. Kako prelazimo na jače oblike umjetne inteligencije, kao što su AGI i ASI, uključivanje više ljudskih ponašanja postaje istaknutije, kao što je sposobnost tumačenja tona i emocija. Umjetna opća inteligencija (AGI) ili opća AI, teorijski je oblik AI gdje bi stroj imao inteligenciju jednaku ljudima; imao bi samosvijest koja ima sposobnost rješavanja problema, učenja i planiranja budućnosti. Umjetna super inteligencija (ASI), također poznata kao superinteligencija, nadmašila bi ljudsku inteligenciju i sposobnost. Chatbotovi i virtualni asistenti, poput Siri i Alexe, zagrebali su površinu ovoga, ali su još uvijek primjeri ANI-a. Još ne postoji niti jedan oblik jake umjetne inteligencije, ali stalna istraživanja u ovom području se nastavljaju. [10, 11]

## 3.1. Strojno učenje

Strojno učenje (engl. machine learning) je grana umjetne inteligencije i računalne znanosti koja se usredotočuje na korištenje podataka i algoritama za oponašanje načina na koji ljudi uče, postupno poboljšavajući njegovu točnost. [12]

Algoritmi strojnog učenja mogu se organizirati u različite kategorije na temelju različitih načela. Na primjer, ovisno o korištenju oznaka uzoraka za trening, mogu se kategorizirati u nadzirano učenje, polunadzirano učenje i algoritme učenja bez nadzora.

U nadziranom učenju svaki uzorak sadrži dva dijela: jedan su ulazna opažanja ili značajke, a drugi su izlazna opažanja ili oznake. Obično su ulazna opažanja uzroci, a izlazna opažanja učinci. Svrha učenja pod nadzorom je izvući funkcionalni odnos iz podataka o obuci koji se dobro generaliziraju na podatke testiranja. Oblik odnosa je skup jednadžbi i brojevanih koeficijenata ili pondera. Nadzirano učenje bavi se dvama glavnim zadacima: regresija (npr. linearna regresija, nelinearna regresija, Bayesova linearna regresija) i klasifikacija (npr. Naivni Bayesov klasifikator, stroj potpornih vektora, logistička regresija). [1]

U nenadziranom učenju imamo samo jedan skup zapažanja i nema informacija o oznaci za svaki uzorak. Obično su ova opažanja ili značajke uzrokovane skupom neopaženih ili latentnih varijabli. Glavna svrha nenadziranog učenja je otkriti odnose između uzoraka ili otkriti latentne varijable iza promatranja. Najčešća tehnika nenadziranog učenja je grupiranje (engl. clustering). [1]

Polunadzirano učenje spada između nadziranog i nenadziranog učenja. Koristi i označene podatke (obično nekoliko) i neobilježene podatke (obično mnogo) tijekom procesa obuke. Algoritmi za polunadzirano učenje razvijeni su uglavnom zato što je označavanje podataka vrlo skupo ili nemoguće u nekim aplikacijama. Primjeri polunadziranog učenja uključuju polunadzirane sustave klasifikacije. [1]

## 3.2. Duboko učenje

Duboko učenje (engl. deep learning) je podskup strojnog učenja koji se temelji na strukturi neuronske mreže koja je inspirirana ljudskim mozgom. Neuronske mreže čine okosnicu algoritama dubokog učenja, a broj slojeva čvorova ili dubina neuronskih mreža razlikuje jednu neuronsku mrežu od algoritma dubokog učenja, koji mora imati više od tri. [11, 13]

Budući da se duboko učenje i strojno učenje obično koriste naizmjenično, vrijedno je napomenuti nijanse između njih. Kao što je gore spomenuto, i duboko učenje i strojno učenje su potpolja umjetne inteligencije, a duboko učenje je zapravo podpodručje strojnog učenja. Način na koji se razlikuju je u tome kako svaki algoritam uči i koliko podataka svaka vrsta algoritma koristi. [10, 11]

Konvencionalni klasifikatori strojnog učenja uglavnom se oslanjaju na unaprijed definirane algoritme s projektiranim značajkama i eksplicitnim parametrima. Ljudski stručnjaci određuju skup značajki kako bi razumjeli razlike između unosa podataka, što obično zahtijeva više strukturiranih podataka za učenje. Takve su značajke dizajnirane za kvantificiranje specifičnih radiografskih karakteristika, kao što su 3D oblik tumora ili intratumorska tekstura i raspodjela intenziteta piksela (histogram). Koriste se kao ulazni podaci za najsuvremenije modele strojnog učenja koji su osposobljeni za razvrstavanje pacijenata na načine koji mogu podržati kliničko donošenje odluka. Iako se takve značajke percipiraju kao diskriminativne, one se oslanjaju na stručnu definiciju i stoga ne predstavljaju nužno najoptimalniji pristup kvantifikaciji obilježja za zadatak diskriminacije. Štoviše, unaprijed definirane značajke često se ne mogu prilagoditi varijacijama u modalitetima snimanja, kao što su računalna tomografija (CT), pozitronska emisijska tomografija (PET) i magnetska rezonanca (MR), te njihove povezane karakteristike signal-šum. Primjeri ovih modela uključuju strojeve potpornih vektora i slučajne šume. [13]

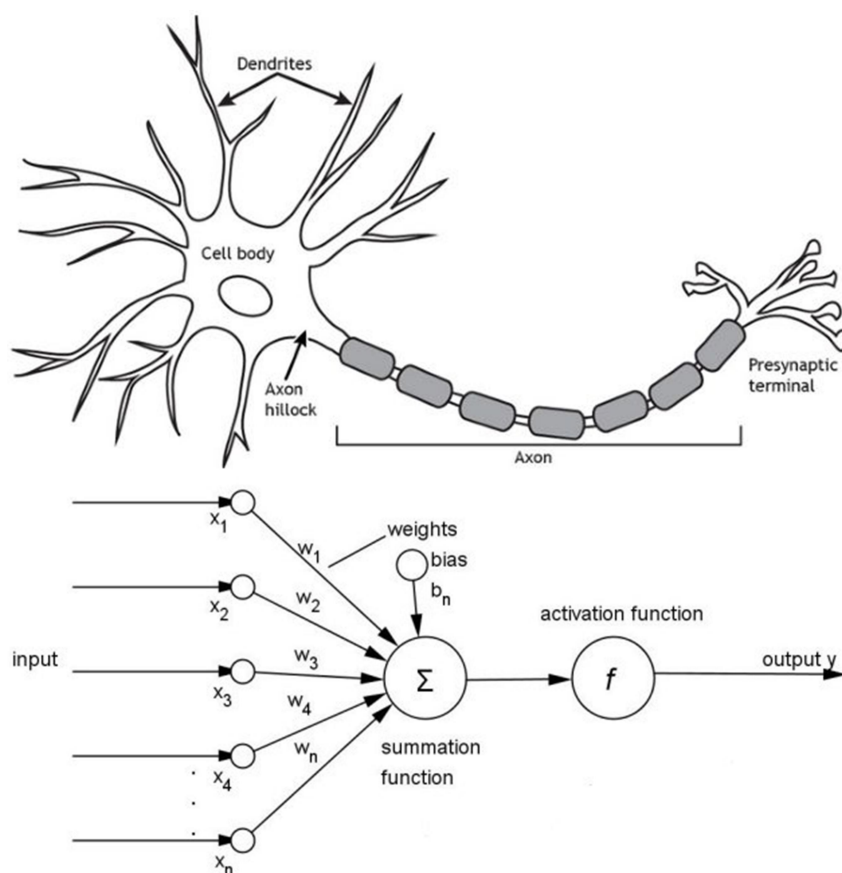
Nedavni napredak u istraživanju umjetne inteligencije doveo je do novih, nedeterminističkih algoritama dubokog učenja koji ne zahtijevaju eksplicitnu definiciju značajki, što predstavlja bitno drugačiju paradigmu u strojnom učenju. Temeljne metode dubokog učenja postoje desetljećima. Međutim, tek je posljednjih godina dostupno dovoljno podataka i računске snage. Bez eksplicitne predefinacije ili odabira značajki od strane ljudskih stručnjaka, ovi algoritmi automatski uče diskriminativne značajke iz podataka, dajući im mogućnost aproksimacije vrlo složenih nelinearnih odnosa i omogućavajući korištenje većih skupova podataka. Ovaj pristup vođen podacima omogućuje apstraktnije definicije značajki, čineći ga informativnijim i generaliziranijim. S obzirom na svoju sposobnost učenja složenih prikaza podataka, također često otporno na neželjene varijacije, kao što je varijabilnost među čitateljima, te se stoga može primijeniti na veliki broj kliničkih stanja i parametara. [13]

"Duboko" strojno učenje može iskoristiti označene skupove podataka, također poznate kao nadzirano učenje, kako bi informirao svoj algoritam, ali ne zahtijeva nužno označeni skup podataka. Može unositi nestrukturirane podatke u sirovom obliku (npr. tekst, slike) i može automatski odrediti skup značajki koje razlikuju različite kategorije podataka jedne od drugih. [12]

Duboko učenje i neuronske mreže prvenstveno su zaslužne za ubrzanje napretka u područjima, kao što su računalni vid, obrada prirodnog jezika i prepoznavanje govora. Na mnogo načina duboko učenje može odražavati ono što rade obučeni radiolozi, odnosno identificirati parametre slike, ali i odmjeriti važnost tih parametara na temelju drugih čimbenika za donošenje kliničke odluke. [12, 13]

### 3.3. Umjetne neuronske mreže

Umjetne neuronske mreže (engl. artificial neural network, ANN) ili simulirane neuronske mreže (SNN), podskup su strojnog učenja i u središtu su algoritama dubokog učenja. Njihovo ime i struktura inspirirani su ljudskim mozgom, oponašajući način na koji biološki neuroni signaliziraju jedni drugima. Sastoje se od slojeva čvorova (umjetnih neurona) koji se dijele na ulazni sloj, jedan ili više skrivenih slojeva i izlazni sloj. [12]



Slika 3: usporedba biološkog neurona i neurona umjetne mreže

Ulazne signale označavamo sa  $x_1, \dots, x_n$ , a težine (engl weights) sa  $w_1, \dots, w_n$ . Ulazni signali općenito su realni brojevi u intervalu  $[-1, 1]$ ,  $[0, 1]$  ili samo elementi iz  $\{0, 1\}$ , kada govorimo o Booleovom ulazu. Svaki čvor, ili umjetni neuron, povezuje se s drugim množenjem ulaznih signala i težina, te se težinskoj sumi dodaje pomak  $b$  (engl. bias). Ako je izlaz bilo kojeg pojedinačnog čvora iznad navedene vrijednosti praga, taj se čvor

aktivira primjenom aktivacijske funkcije i šalje podatke sljedećem sloju mreže. Inače se podaci ne prosljeđuju na sljedeći sloj mreže. "Duboko" u dubokom učenju odnosi se na dubinu slojeva u neuronskoj mreži. Neuronska mreža koja se sastoji od više od tri sloja (koji bi uključivali ulaze i izlaz) može se smatrati algoritmom dubokog učenja ili dubokom neuronskom mrežom. Neuronska mreža koja ima samo dva ili tri sloja je samo osnovna neuronska mreža. [14, 15]

### **3.3.1. Vrste umjetne neuronskih mreža**

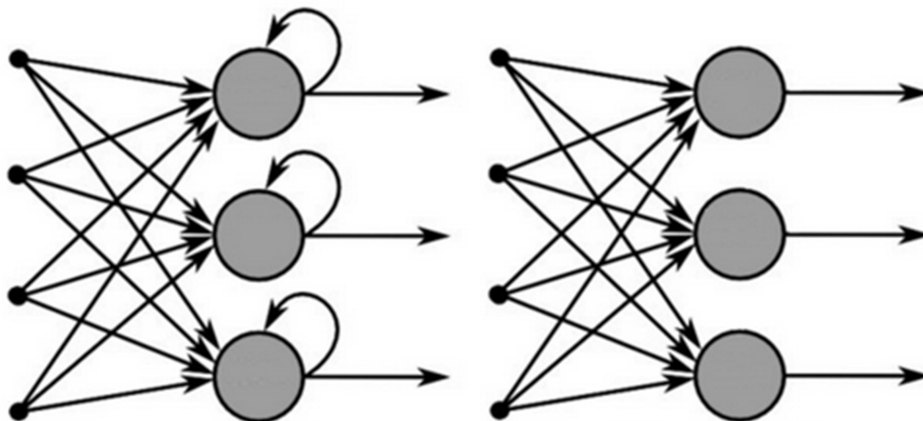
Postoje različite vrste neuronskih mreža koje se koriste za različite slučajeve upotrebe i vrste podataka. U nastavku su navedene najpoznatije vrste neuronskih mreža.

Perceptron je najstarija i najjednostavnija neuronska mreža koju je stvorio Frank Rosenblatt 1958. Sastoji se od neurona koji se nazivaju Linear Threshold Unit (LTU) i može se trenirati s jednoslojnom arhitekturom, koja se sastoji od ulaznog i izlaznog sloja. U ulaznom sloju se ne vrše nikakva računanja, stoga se ne ubraja u slojeve mreže. Perceptron se smatra linearnim modelom, koji zabranjuje njegovu primjenu za zadatke koji uključuju komplicirane uzorke podataka, unatoč korištenju nelinearnih aktivacijskih funkcija u izlaznom sloju. Takvo ograničenje uspješno se zaobilazi uvođenjem takozvanog „skrivenog“ sloja između ulaznog i izlaznog sloja s tim da su jedinice susjednih slojeva u potpunosti povezane jedna s drugom, ali ne postoje veze među jedinicama u istom sloju. Takva mreža naziva se višeslojni perceptron (engl. multi-layer perceptrons). Pod blagom pretpostavkom o funkciji aktivacije, dvoslojna neuronska mreža s konačnim brojem skrivenih jedinica može aproksimirati bilo koju kontinuiranu funkciju, te se stoga smatra univerzalnim aproksimatorom. Međutim, također je moguće aproksimirati složene funkcije na istu točnost koristeći „duboku“ arhitekturu, tj. više od dva sloja, s ukupno mnogo manjim brojem jedinica. Stoga je moguće smanjiti broj parametara koji se mogu trenirati, čime se omogućuje vježbanje s relativno malim skupom podataka. [15, 21]

Acikličke mreže (engl. feedforward) sastoje se od ulaznog sloja, skrivenog sloja ili slojeva i izlaznog sloja. Nema povratnih veza između neurona pa signali koji krenu od ulaznih neurona nakon određenog broja prijelaza dolaze do izlaza mreže, tj. propagacija signala je jednosmjerna. Iako se te neuronske mreže također obično nazivaju višeslojni perceptroni, važno je napomenuti da se zapravo sastoje od sigmoidnih neurona, a ne perceptrona, budući da je većina problema u stvarnom svijetu nelinearna. Podaci se obično unose u ove modele kako bi ih trenirali, a oni su temelj za računalni vid, obradu prirodnog jezika i druge neuronske mreže. [14, 15]

Rekurentne neuronske mreže (RNN) poznate su po njihovim povratnim petljama. Vrsta umjetne neuronske mreže koja koristi sekvencijalne podatke ili podatke vremenske serije. Ovi algoritmi dubokog učenja obično se koriste za redovne ili vremenske probleme, kao što su predviđanje budućih ishoda (predviđanja tržišta dionica ili predviđanje prodaje), prijevod jezika, obrada prirodnog jezika (engl. natural language processing, NLP), prepoznavanje govora. Ugrađeni su u popularne aplikacije kao što su Siri, glasovno pretraživanje i Google Translate. Kao i feedforward mreže i konvolucijske neuronske mreže (CNN), rekurentne neuronske mreže koriste podatke treninga za učenje. Odlikuju se svojim "pamćenjem" jer uzimaju informacije iz prethodnih ulaza kako bi utjecali na trenutni ulaz i izlaz. Dok tradicionalne duboke neuronske mreže pretpostavljaju da su ulazi i izlazi neovisni jedan o drugom, izlaz rekurentnih neuronskih mreža ovisi o prethodnim elementima unutar niza. Dok bi budući događaji također bili od pomoći u određivanju izlaza danog niza, jednosmjerne rekurentne neuronske mreže ne mogu uzeti u obzir te događaje u svojim predviđanjima. Varijante mrežne arhitekture RNN-ova su: dvosmjerne rekurentne neuronske mreže (engl. bidirectional recurrent neural networks, BRNN), popularna RNN arhitektura dugotrajna kratkoročna memorija (engl. long short-term memory, LSTM) i, slične LSTM-ovima, zatvorene rekurentne jedinice (engl. gated recurrent units). [16]





Slika 4: rekurentna mreža (lijevo) i aciklička mreža (desno)

Konvolucijske neuronske mreže (engl. convolutional neural network, CNN) slične su mrežama za prijenos podataka, ali se obično koriste za prepoznavanje slika, uzoraka i/ili računalni vid. Prije CNN-a, za identifikaciju objekata na slikama korištene su ručne, dugotrajne metode ekstrakcije značajki. Međutim, konvolucijske neuronske mreže sada pružaju skalabilniji pristup klasifikaciji slika i zadacima prepoznavanja objekata, koristeći principe iz linearne algebre, točnije množenje matrica, za prepoznavanje uzoraka unutar slike. Međutim, oni mogu biti računski zahtjevni, zahtijevajući grafičke procesorske jedinice (GPU) za obuku modela. [17]

### 3.4. Primjeri kliničkih područja primjene umjetne inteligencije u radiologiji

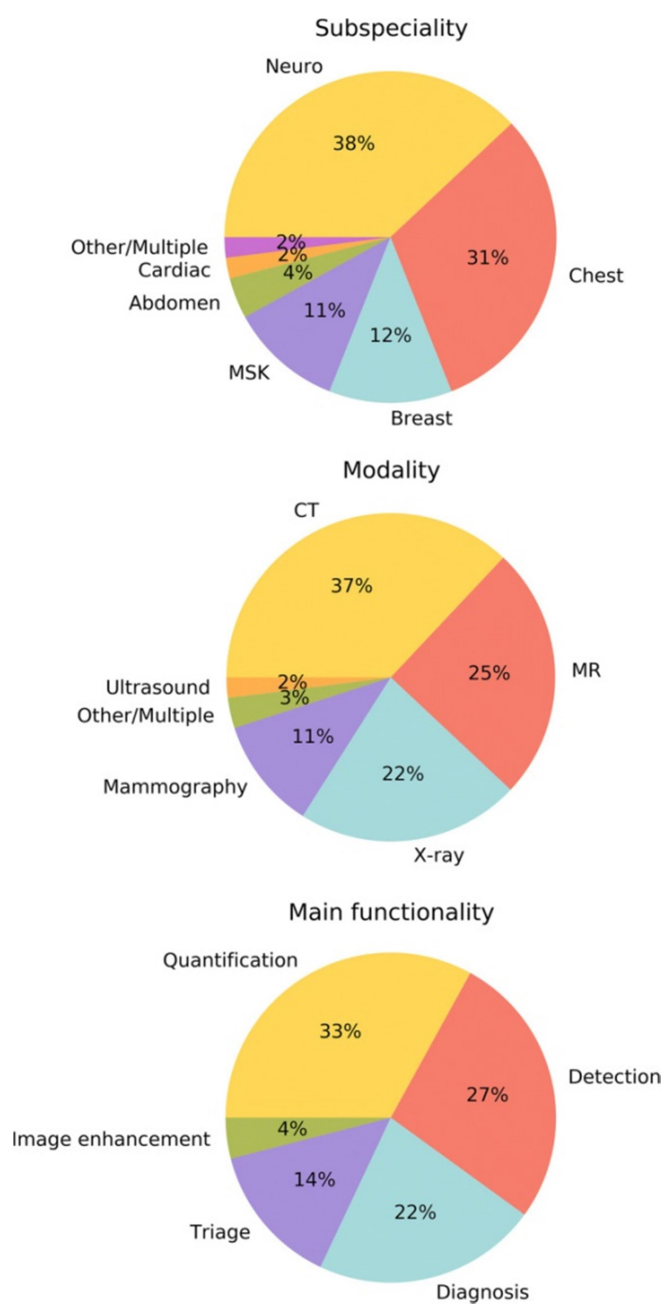
Radiolozi vizualno pregledavaju hrpu slika dok povremeno prilagođavaju ravnine gledanja te postavke širine i razine prozora. Oslanjajući se na obrazovanje, iskustvo, razumijevanje zdrave radiografije te poznavanje osnovne fiziologije bolesti, radiolozi su osposobljeni za prepoznavanje abnormalnosti na temelju promjena u intenzitetu snimanja ili pojave neobičnih uzoraka. Ovi kriteriji, i mnogi drugi, spadaju u donekle subjektivnu matricu odluka koja omogućuje razmišljanje o problemima u rasponu od otkrivanja plućnih nodula do lezija dojke i polipa debelog crijeva. [18, 19]

Noviji modaliteti snimanja kao što su CT i MR mogu pružiti detaljnije informacije s tanjim slikama i/ili više serija slika, a vrijeme potrebno za prikupljanje tih slika je kraće nego prije. Stoga se broj snimaka prikupljenih u svakom pregledu povećava, dok broj radiologa koji tumače te slike ne. [20]

To je pridonijelo dramatičnom povećanju radnog opterećenja radiologa. Kako radiologija uključuje vizualnu percepciju, kao i donošenje odluka pod nesigurnošću, pogreške su neizbježne - osobito u takvim ograničenim uvjetima. [18, 19]

Za razliku od takvog kvalitativnog zaključivanja, umjetna inteligencija izvrsno prepoznaje složene obrasce u slikovnim podacima i može dati kvantitativnu procjenu na automatiziran način. Primarni pokretač pojave umjetne inteligencije u medicinskom oslikavanju bila je želja za većom djelotvornošću i učinkovitošću kliničke skrbi. Besprijekorno integrirana AI komponenta unutar radnog tijeka snimanja povećala bi učinkovitost, smanjila pogreške i postigla ciljeve uz minimalan ručni unos pružajući obučanim radiolozima unaprijed pregledane slike i identificirane značajke. Gotovo svi radiološki zadaci temeljeni na slici ovise o kvantificiranju i procjeni radiografskih karakteristika slika. Ove karakteristike mogu biti važne za klinički zadatak koji se radi, odnosno za otkrivanje, karakterizaciju ili praćenje bolesti. [18]

Rezultati studije (slika 5) pokazuju da se dostupni softver za umjetnu inteligenciju uglavnom bavi neuroradiologijom (38%) i radiologijom prsnog koša (31%), a zatim radiologijom dojke (12%) i mišićno-koštanog sustava (11%). Što se tiče modaliteta, proizvodi su raspoređeni na CT (37%), MR (25%) i radiografiju (22%). Glavni zadaci koje izvršava softver su zadaci kvantifikacije (33%), poput segmentacije regije ili izvođenja automatiziranih mjerenja. [18]



Slika 5: rezultati studije: udio regije interesa, modaliteta snimanja i vrste AI zadatka

U nastavku su predstavljane neke tipične primjene umjetne inteligencije u radiologiji u različitim modalitetima koje mogu pomoći u otkrivanju, dijagnozi, postavljanju i klasifikaciji stanja. Podijeljene su u sljedeće kategorije: klasifikacija, segmentacija, detekcija i druge. [20, 21]

### 3.4.1. Klasifikacija

U analizi medicinske slike, klasifikacija s dubokim učenjem obično koristi ciljane lezije prikazane na medicinskim slikama, a te se lezije razvrstavaju u dvije ili više klasa. Ovi se zadaci postižu kvantificiranjem radioloških karakteristika abnormalnosti, kao što su veličina, opseg i unutarnja tekstura. Na primjer, duboko učenje se često koristi za klasifikaciju plućnih nodula na slikama kompjuterizirane tomografije (CT) kao benignih ili malignih. Automatizacija putem umjetne inteligencije može, u načelu, uzeti u obzir velik broj kvantitativnih značajki zajedno s njihovim stupnjevima relevantnosti, dok svaki put izvršava zadatak na reproducibilan način. Potrebno je pripremiti veliki broj podataka za obuku s odgovarajućim oznakama za učinkovitu klasifikaciju pomoću CNN-a. Za klasifikaciju plućnih nodula, CT slike plućnih nodula i njihovih oznaka (tj. benignih ili malignih) koriste se kao podaci za trening. AI može automatski identificirati ove značajke, isto tako benigne i maligne lezije jetre na snimkama kompjutorizirane tomografije (CT) i magnetske rezonance (MR), kao i mnoge druge, dok ih tretira kao slikovne biomarkere. Takvi bi se biomarkeri stoga mogli koristiti za predviđanje vjerojatnosti maligniteta među ostalim kliničkim krajnjim točkama uključujući procjenu rizika, diferencijalnu dijagnozu, prognozu i odgovor na liječenje. [18, 21]

### 3.4.2. Segmentacija

Medicinske slike sadrže mnoge strukture uključujući normalne strukture kao što su organi, kosti, mišići, masnoće te abnormalne strukture kao što su tumori i prijelomi. Segmentacija je proces identifikacije struktura na slikama. Segmentiranje struktura, normalnih i abnormalnih, iz medicinskih slika nije trivijalno zbog složenosti i varijabilnosti regije od interesa. Primjeri problema koji kompliciraju segmentaciju medicinske slike uključuju normalne anatomske varijacije, postkirurške anatomske varijacije, nejasne i nepotpune granice, neodgovarajući kontrast, artefakte i šum. U kliničkoj radijacijskoj onkologiji, opseg i tumorskog i netumorskog tkiva mora biti precizno segmentiran za planiranje liječenja zračenjem i optimizacije doze zračenja. [1, 18]

Segmentaciju mogu ručno izvesti primjerice radiolozi, što je dugotrajan proces. Međutim, CNN se također može primijeniti na ovaj zadatak. U većini slučajeva, sustav segmentacije izravno prima cijelu sliku i daje rezultat segmentacije. Podaci za treniranje za sustav segmentacije sastoje se od medicinskih slika koje sadrže organ ili strukturu od interesa i rezultat segmentacije koji se uglavnom dobiva iz prethodno provedene ručne segmentacije. [21]

Postoje arhitekture, kao što je U-net, posebno dizajnirane za medicinske slike. Studije su izvijestile da je jedan sustav dubokog učenja sposoban izvoditi različite zadatke segmentacije na više modaliteta i tipova tkiva, uključujući MR mozga, MR dojke i CT angiografiju srca (CTA), bez obuke za specifične zadatke. [18]

### **3.4.3. Detekcija**

Uobičajeni zadatak radiologa je otkrivanje abnormalnosti unutar medicinskih slika. Abnormalnosti mogu biti rijetke i moraju se detektirati među mnogim normalnim slučajevima. Umor uzrokovan povećanim opterećenjem pregledavanja većeg broja slika po pregledu, u kombinaciji s pritiscima na vremensku učinkovitost, može rezultirati lažno negativnim nalazom (npr. promašenim plućnim čvorovima). Kako bi pomogli liječnicima u tumačenju medicinskih slika i kako bi se smanjila lažno negativna očitavanja, računalno potpomognuto otkrivanje (CADe) i računalno potpomognuta dijagnoza (CADx) pružaju učinkovit način za smanjenje vremena čitanja, povećanje osjetljivosti detekcije i poboljšanje točnosti dijagnoze. Radiolog prvo pregledava radiološke snimke, zatim aktivira CAD softver za prepoznavanje uzoraka koji identificira sumnjive značajke na slici i skreće pozornost radiologu koji ponovno procjenjuje CAD označena područja prije izdavanja konačnog izvješća. [1]

Korištenjem modela konvolucijskih neuronskih mreža (CNN) u CADe rezultati nekoliko studija pokazuju da je postigao slične performanse u usporedbi s ljudskim čitačima. Područja primjene uključuju otkrivanje lezija, odnosno mikrokalcifikata i masa na mamografima, karcinom prostate u multiparametrijskom MR, plućne embolije na CT angiografiji i tuberkuloze na radiografiji prsnog koša. [18, 21]

Osim gore navedenih, valja spomenuti još nekoliko primjena umjetne inteligencije u radiologiji.

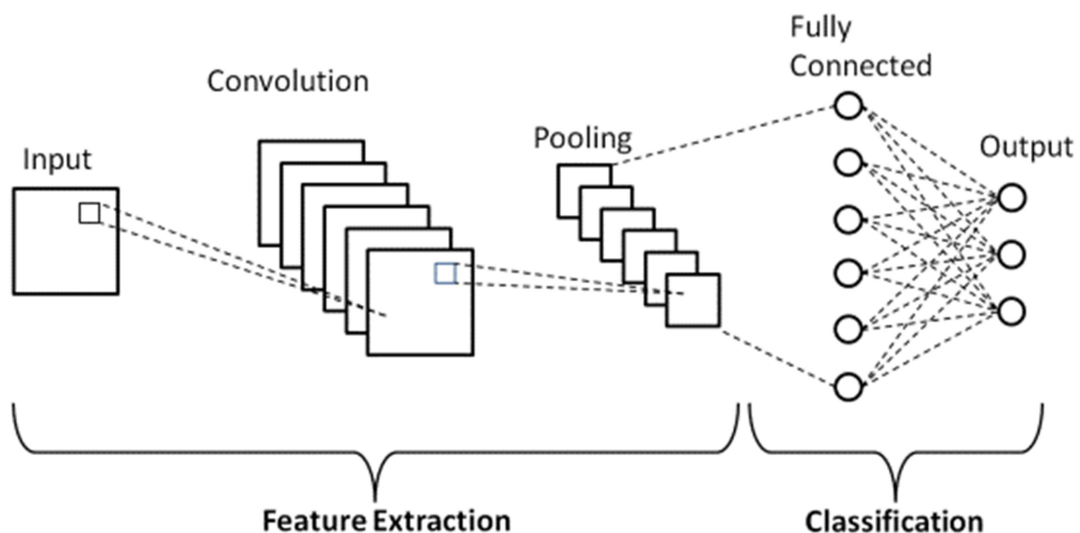
Praćenje bolesti ključno je za dijagnozu, kao i za procjenu odgovora na liječenje ili uspjeha terapije. Tijek rada uključuje predproces registracije slike gdje se oboljelo tkivo poravnava na višestrukim skeniranjima, nakon čega slijedi evaluacija jednostavnih metričkih podataka o njima pomoću unaprijed definiranih protokola. Slijedi jednostavan protokol usporedbe podataka koji se koristi za kvantificiranje promjena. U onkologiji, na primjer, ovi protokoli definiraju informacije o veličini tumora. Arhitekture dubokog učenja, kao što su rekurentne neuronske mreže, vrlo su prikladne za takve formate podataka vremenskog niza i očekuje se da će pronaći široku primjenu u zadacima praćenja. [18]

Radiolozi rutinski sastavljaju svoja izvješća kao rezultate interpretacije medicinskih slika. Budući da sažimaju medicinske slike kao tekstualne podatke u izvješćima, moglo bi biti moguće prikupiti korisne informacije o dijagnozi bolesti za automatsko generiranje izvješća pomoću vjerojatnostnog modela koji se temelji na neuronskoj mreži. [19, 21]

Algoritmi dubokog učenja se poboljšavaju s podacima, odnosno, kako se svaki dan generira sve više podataka i uz stalne istraživačke napore, vidljiva su poboljšanja u izvedbi. Svi ti napretci obećavaju povećanu točnost i smanjenje broja rutinskih zadataka koji iscrpljuju vrijeme i trud. [18]

## 4. Konvolucijska neuronska mreža

Konvolucijska neuronska mreža (CNN), klasa umjetnih neuronskih mreža koja je postala dominantna u različitim zadacima računalnog vida, privlači interes u raznim domenama, uključujući radiologiju. CNN je vrsta modela dubokog učenja za obradu podataka koji imaju rešetkasti uzorak (engl. grid-like topology), kao što su slike, koji je inspiriran organizacijom vizualnog korteksa mačke i dizajniran tako da automatski i prilagodljivo uči prostorne hijerarhije značajki kroz propagaciju unatrag korištenjem višestrukih građevnih blokova, kao što su slojevi konvolucije, slojevi sažimanja i potpuno povezani slojevi. Konvolucijski sloj je prvi sloj konvolucijske mreže. Konvolucijski sloj igra ključnu ulogu u CNN-u, koji se sastoji od niza matematičkih operacija, kao što je konvolucija, specijalizirana vrsta linearne operacije. Dok konvolucijski slojevi mogu biti praćeni dodatnim konvolucijskim slojevima ili slojevima sažimanja, potpuno povezani sloj je završni sloj. Izlaz prethodnog sloja dovodi se na ulaz sljedećeg sloja te su ulazi i izlazi neurona povezani između dva sloja. Sa svakim slojem, CNN povećava svoju složenost, identificirajući veće dijelove slike. Prvi slojevi usredotočuju se na jednostavne značajke, kao što su boje i rubovi. Kako slikovni podaci napreduju kroz slojeve CNN-a, on počinje prepoznavati veće elemente ili oblike objekta dok konačno ne identificira željeni objekt. [21, 22]



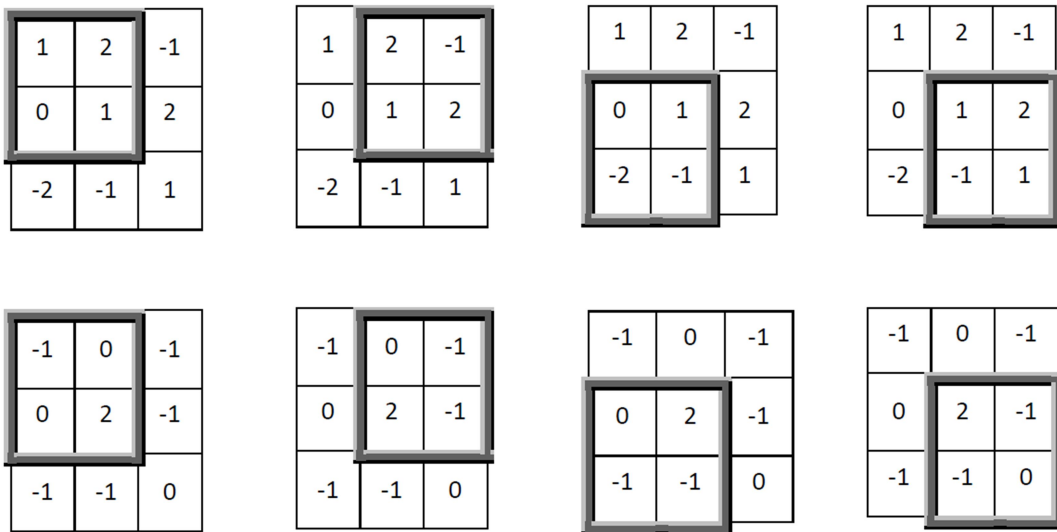
Slika 6: arhitektura konvolucijske neuronske mreže

## 4.1. Konvolucijski sloj

Konvolucijski sloj je temeljni građevni blok CNN-a i u njemu se događa većina izračunavanja. Sadrži nekoliko komponenti, a to su ulazni podaci, filter i mapa značajki. Ako je ulaz slika u boji, koja se sastoji od matrice piksela u 3D, ulaz će imati tri dimenzije – visinu, širinu i dubinu – koje odgovaraju RGB-u na slici. Proces poznat kao konvolucija specijalizirana je vrsta linearne operacije koji se koristi za ekstrakciju značajke, gdje se mali niz brojeva, kernel ili filter, sa težinama primjenjuje na ulaz, a to je niz brojeva koji se naziva tenzor. Filter se primjenjuje na područje slike, a između svakog elementa filtera i ulaznog tenzora izračunava se umnožak koji se zatim unosi u izlazni niz. Nakon toga, filter se pomiče za korak, ponavljajući postupak sve dok filter ne pređe preko cijele slike. Konačni izlaz, dobivene izlazne vrijednosti u odgovarajućem položaju izlaznog tenzora, je dvodimenzionalna mapa značajki. Dva ključna hiperparametra koja definiraju operaciju konvolucije su veličina i broj filtera. Veličina filtera je obično matrica  $3 \times 3$ , ali može biti  $5 \times 5$  ili  $7 \times 7$ . Broj filtera je proizvoljan i određuje dubinu izlaznih mapa značajki. Na primjer, tri različita filtera dala bi tri različite mape značajki, stvarajući dubinu od tri. [21, 22]

Pomak (engl. stride) je broj piksela na kojima se filter pomiče preko ulazne matrice, to jest označava za koliko će se filter pomicati po širini, odnosno visini tijekom procesa konvolucije. Uobičajeni izbor koraka je jedan, dok je vrijednost koraka dva ili više rijetka. Veći korak daje manji učinak. Ponekad se koristi kako bi se postiglo smanjenje uzorkovanja (engl. downsampling) mapa značajki. [21]



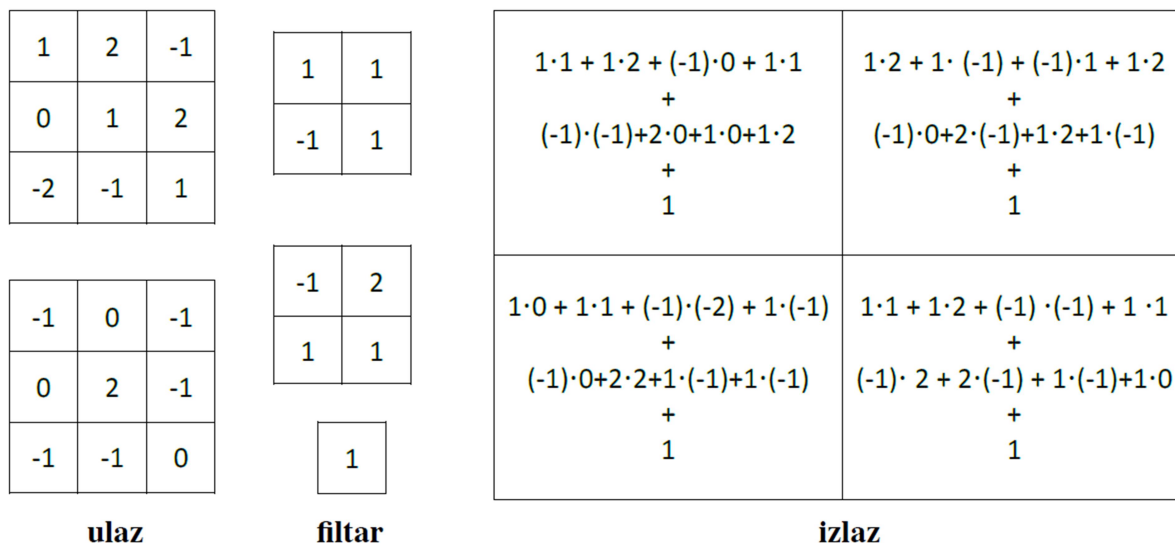


Slika 7: primjer prelaženja filtera po slici s pomaknom veličine jedan

Kako bi izračunali izlaz konvolucijskog sloja, odnosno odredili aktivacijsku mapu, potrebno je provesti konvoluciju nad ulazom i filterom. Konvolucija se provodi tako da odgovarajuće ulaze pomnožimo s težinama, dobivene vrijednosti potrebno je sumirati i dodati prag. Potrebno je naglasiti da pri izračunu vrijednosti za jedno polje potrebno je sumirati umnoške ulaza i težina svih dubina. Za izračun vrijednosti sljedećeg polja potrebno je pomjeriti filter za definirani pomak i ponoviti postupak. [25]

Za definiranje točne veličine izlaza konvolucijskog sloja. Širina izlaza označavat će se s  $W_y$ , a širina ulaza s  $W_x$ . Korak pomaka filtra označavat će se sa  $S$ , dok će se s oznakom  $F$  označavati veličina (kvadratnog) filtera. [25] Širina izlaza definirana je izrazom:

$$W_y = (W_x - F) / S + 1$$



Slika 8: primjer rada konvolucijskog sloja

Širina izlazne mape za primjer sa slike može se također izračunati prema izrazu, uz veličinu kvadratnog filtra,  $F = 2$ , i korakom  $S = 1$ :

$$W_y = (W_x - F) / S + 1 = (3 - 2) / 1 + 1 = 2$$

Operaciju konvolucije najlakše je opisati algebarski, stoga je donjim izrazom opisana veza između ulaza i izlaza. Ulaz je zapravo izlaz iz prethodnog sloja pa je oznađen s  $y^{l-1}_d(i, j)$ , gdje i i j određuju poziciju unutar matrice, d označava dubinu, dok l označava o kojem sloju se radi. Težine unutar filtra označene su oznakom w. Kao i u ranijim primjerima, veličina filtra označena je s F, funkcija f predstavlja aktivacijsku funkciju, dok b označava prag. [25]

$$y^l(i, j) = \sum_{a=0}^{F-1} \sum_{b=0}^{F-1} \sum_{d=0}^{D-1} f(w_d(i, j) \cdot y^{l-1}_d(i + a, j + b) + b)$$

Ključna značajka operacije konvolucije je dijeljenje težine: filteri se dijele na svim pozicijama slike. Proces obuke CNN modela s obzirom na sloj konvolucije je identificirati filtere koji najbolje rade za dani zadatak na temelju zadanog skupa podataka za obuku. Filteri su jedini parametri koji se automatski uče tijekom procesa treniranja u sloju konvolucije. S druge strane, veličina i broj filtera, nadopunjavanje i pomak su hiperparametri koje je potrebno postaviti prije početka procesa treniranja. [21]

### 4.1.1. Nadopunjavanje

Ako se radi o slici dimenzija  $n \times n$  koja se konvoluirala s  $f \times f$  filterom, dimenzije izlazne slike su  $(n - f + 1) \times (n - f + 1)$ . Iz ovog proizlaze dva problema. Prvo, svaki put kada se obavi konvolucija slika se smanjuje. Drugo, ako se uzme piksel na rubu slike, taj se piksel samo jednom koristi u izlaznim podacima, to jest izlaznoj slici. Ako se uzme piksel u sredini, onda se taj piksel koristi više puta u izlaznim podacima. Iz navedenog se može zaključiti da se na izlaznoj slici gubi informacija o pikselima koji se nalaze na rubovima. Kako se ulazna slika ne bi smanjila, i kako se ne bi izgubila informacija o podacima koji se nalaze na rubovima slike, koristi se nadopunjavanje. [24]

Nadopunjavanje, obično nulama (engl. zero padding), je tehnika gdje se redovi i stupci nula dodaju sa svake strane ulaznog tenzora. Moderne arhitekture CNN-a najčešće koriste popunjavanje nulama, u suprotnom bi svaka uzastopna mapa značajki postala bi manja nakon operacije konvolucije. [21] Na sljedećoj slici je prikazana operacija nadopunjavanja nulama izvan rubova slike.

|   |     |     |     |     |     |   |
|---|-----|-----|-----|-----|-----|---|
| 0 | 0   | 0   | 0   | 0   | 0   | 0 |
| 0 | 60  | 113 | 56  | 139 | 85  | 0 |
| 0 | 73  | 121 | 54  | 84  | 128 | 0 |
| 0 | 131 | 99  | 70  | 129 | 127 | 0 |
| 0 | 80  | 57  | 115 | 69  | 134 | 0 |
| 0 | 104 | 126 | 123 | 95  | 130 | 0 |
| 0 | 0   | 0   | 0   | 0   | 0   | 0 |

Kernel

|    |    |    |
|----|----|----|
| 0  | -1 | 0  |
| -1 | 5  | -1 |
| 0  | -1 | 0  |

|     |      |     |      |     |
|-----|------|-----|------|-----|
| 114 | 328  | -26 | 470  | 158 |
| 53  | 266  | -61 | -30  | 344 |
| 403 | 116  | -47 | 295  | 244 |
| 108 | -135 | 256 | -128 | 344 |
| 314 | 346  | 279 | 153  | 421 |

Slika 9: primjer nadopunjavanja nulama

Općenito, kada se koristi nadopunjavanje (p), dimenzije izlazne slike će biti

$$(n + 2p - f + 1) \times (n + 2p - f + 1)$$

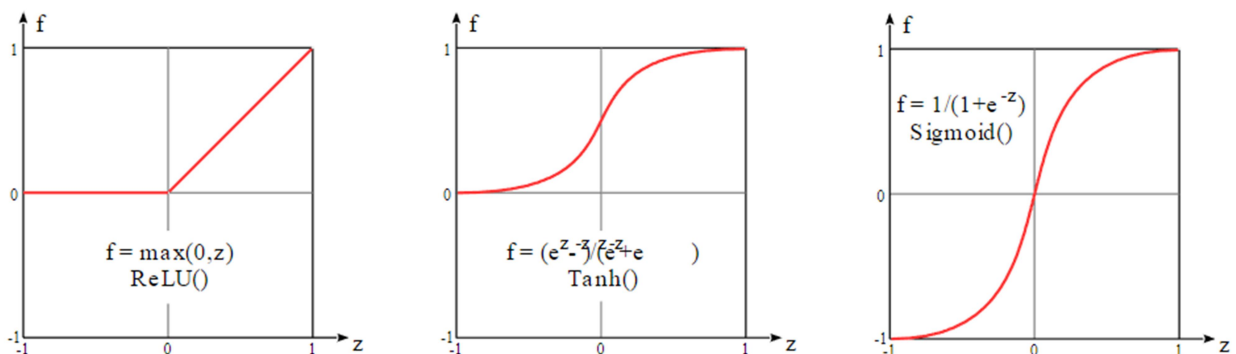
Dvije najčešće korištene metode nadopunjavanja su takozvane valid i same. Ako se koristi valid nadopunjavanje, slika se ne nadopunjuje, dakle izlazna dimenzija slika će biti dimenzija  $(n - f + 1) \times (n - f + 1)$ . Ako se koristi same nadopunjavanje, dimenzije izlazne slike bit će jednake dimenzijama ulazne. [24]

## 4.1.2. Aktivacijska funkcija

Izlazi linearne operacije konvolucije propuštaju se kroz nelinearnu aktivacijsku funkciju. Odabir aktivacijske funkcije ovisi o tipu problema koji se rješava te arhitekturi same neuronske mreže. Iako su glatke nelinearne funkcije, kao što je sigmoidna ili hiperbolička tangenta (tanh) funkcija, ranije korištene jer su matematički prikazi ponašanja biološkog neurona, najčešća nelinearna aktivacijska funkcija koja se trenutno koristi je ispravljena linearna jedinica (engl. rectified linear unit, ReLU), koja jednostavno izračunava funkciju [21]:

$$f(z) = \max(0, z)$$

Grafovi spomenutih funkcija prikazani su na slici 9.



Slika 10: grafovi aktivacijskih funkcija

Kod ReLU aktivacijske funkcije za ulaznu vrijednost  $z > 0$  izlazna vrijednost je  $z$ , a za ulaznu vrijednost  $z < 0$ , izlazna vrijednost je 0. Ova funkcija izbjegava i ispravlja problem nestajanja gradijenata.

Izlazna vrijednost sigmoidne funkcije je uvijek u intervalu  $[0, 1]$ . Krivulja funkcije ima S-oblik. Omogućuje svojstvo nelinearnosti – njezine su kompozicije također nelinearne. Obično se označava s  $\sigma$  i koristi za binarnu klasifikaciju, jer su izlazne vrijednosti vjerojatnosti  $[0, 1]$ .

Izlazne vrijednosti tangens hiperbolne funkcije su na intervalu  $[-1, 1]$ . Tanh je također kao logistička sigmoida, ali bolja. Prednost je u tome što će negativni ulazi biti mapirani jako negativno, a nulti ulazi bit će mapirani blizu nule na tanh grafu.

Funkcija softmax je općenitija logistička aktivacijska funkcija koja se koristi za višeklasnu klasifikaciju. [26]

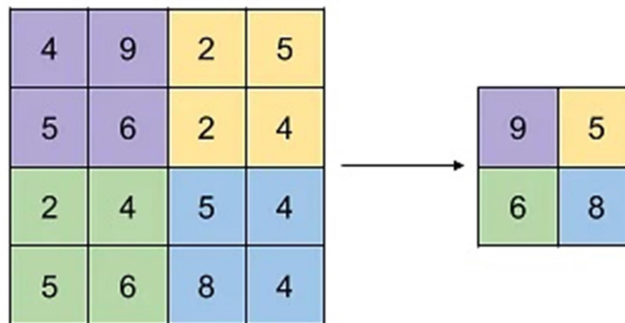
## 4.2. Sloj sažimanja

Sažimanje slojeva, također poznato kao downsampling, pruža tipičnu operaciju smanjenja uzorkovanja koja smanjuje dimenzionalnost u ravnini mapa značajki kako bi se uvela prostorna invarijantnost odnosno neosjetljivost na manje pomake i izobličenja, te smanjio broj sljedećih parametara u ulazu koji se mogu naučiti čime se dobiva na brzini. Važno je napomenuti da ne postoji parametar koji se može naučiti ni u jednom od slojeva sažimanja (engl. pooling layers), dok su veličina filtera, korak i padding hiperparametri u operacijama sažimanja. Slično konvolucijskom sloju, operacija sažimanja pomiče filter preko cijelog ulaza, ali razlika je u tome što ovaj filter nema nikakve težine. Umjesto toga, filter primjenjuje funkciju agregacije na vrijednosti unutar receptivnog polja, popunjavajući izlazni niz. Postoje dvije glavne vrste sažimanja, maksimalnom vrijednošću i prosječnom vrijednošću. [21, 22]

### 4.2.1. Sažimanje maksimalnom vrijednošću

Najpopularniji oblik operacije sažimanja je sažimanje maksimalnom vrijednošću (engl. max pooling). Kako se filter kreće po ulazu, odabire piksel s maksimalnom vrijednošću za slanje u izlazni niz i odbacuje sve ostale vrijednosti. Primjerice, sažimanje maksimalnom vrijednošću s filterom veličine  $2 \times 2$  s korakom 2 smanjuje uzorkovanje u dimenziji mapa značajki za faktor 2. Za razliku od visine i širine, dimenzija dubine mapa značajki ostaje nepromijenjena. [21]

### *Max Pooling*

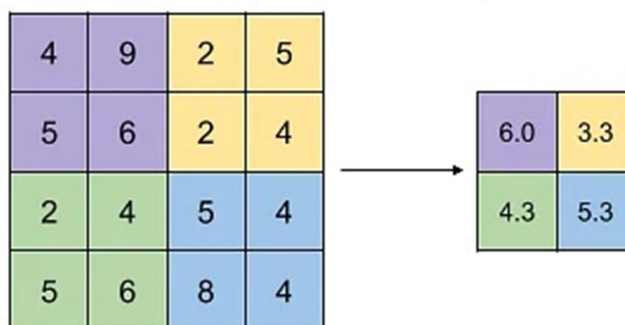


Slika 11: primjer sloja sažimanja maksimalnom vrijednošću

### 4.2.2. Sažimanje srednjom vrijednošću

Za sažimanje srednjom vrijednošću, kako se filter kreće po ulazu, izračunava prosječnu vrijednost unutar receptivnog polja za slanje u izlazni niz. Izvodi ekstremnu vrstu smanjenja uzorkovanja, gdje se karta značajki s veličinom visine  $\times$  širine smanjuje u niz  $1 \times 1$  jednostavnim uzimanjem prosjeka svih elemenata u svakoj mapi značajki, dok je dubina mapa značajki zadržana. Ova se operacija obično primjenjuje samo jednom prije potpuno povezanih slojeva. [21]

### *Avg Pooling*



Slika 12: primjer sloja sažimanja prosječnom vrijednošću

Iako se puno informacija gubi u sloju sažimanja, on također ima niz prednosti za CNN. Pomaže u smanjenju složenosti, poboljšanju učinkovitosti i ograničavanju rizika od prenaučivosti (engl. overfitting), smanjuje broj parametara koji se mogu naučiti i omogućuje CNN-u da prihvati ulaze promjenjive veličine. [21, 22]

### 4.3. Potpuno povezani sloj

Vrijednosti piksela ulazne slike nisu izravno povezane s izlaznim slojem u djelomično povezanim slojevima, odnosno slojevima konvolucije i sažimanja. Međutim, u potpuno povezanom sloju, svaki čvor u izlaznom sloju povezuje se izravno s čvorom u prethodnom sloju. Mape izlaznih značajki završne konvolucije ili sloja sažimanja obično su spljoštene, tj. transformirane u jednodimenzionalni (1D) niz brojeva (ili vektora) i povezane s jednim ili više potpuno povezanih slojeva, također poznatih kao gusti slojevi, u kojem je svaki ulaz povezan sa svakim izlazom pomoću težine koja se može naučiti. Nakon što se kreiraju značajke koje su ekstrahirane konvolucijskim slojevima i smanjene uzorkovanjem slojevima sažimanja, mapiraju se podskupom potpuno povezanih slojeva na konačne izlaze mreže, kao što su vjerojatnosti za svaku klasu u zadacima klasifikacije. Konačni potpuno povezani sloj ima isti broj izlaznih čvorova kao i broj klasa. [21, 22]

Funkcija aktivacije primijenjena na zadnji potpuno povezani sloj obično se razlikuje od ostalih. Za svaki zadatak potrebno je odabrati odgovarajuću funkciju aktivacije. Dok konvolucijski slojevi i slojevi sažimanja imaju tendenciju koristiti ReLu funkcije, potpuno povezani slojevi obično koriste softmax aktivacijsku funkciju za odgovarajuću klasifikaciju ulaza. [21, 22]



## 4.4. Treniranje mreže

Podaci i oznake temeljne istine najvažnije su komponente u istraživanju primjenom dubokog učenja ili drugih metoda strojnog učenja. Pažljivo prikupljanje podataka i oznaka temeljne istine za treniranje i testiranje modela obvezno je za uspješan projekt dubokog učenja, ali dobivanje visokokvalitetnih označenih podataka može biti skupo i dugotrajno. Iako postoji više skupova medicinskih slika otvorenih za javnost, u tim slučajevima treba obratiti posebnu pozornost na kvalitetu temeljnih oznaka temeljne istine. [21]

Podaci su obično podijeljeni u tri skupa: trening, provjera valjanosti (validacijski) i testni skup, iako postoje neke varijante, kao što je unakrsna provjera valjanosti. Skup za trening koristi se za treniranje mreže, gdje se vrijednosti gubitaka izračunavaju putem širenja prema naprijed, a parametri koji se mogu naučiti ažuriraju se putem propagacije unatrag. Skup za provjeru valjanosti koristi se za evaluaciju modela tijekom procesa treninga, fino podešavanje hiperparametara i izvođenje odabira modela. Testni se skup idealno koristi samo jednom na samom kraju projekta kako bi se procijenila izvedba konačnog modela koji je fino podešen i odabran u procesu obuke sa skupovima za treniranje i validaciju. [21]

Potrebni su zasebni skupovi za provjeru valjanosti i testiranja jer obuka modela uvijek uključuje fino podešavanje njegovih hiperparametara i izvođenje odabira modela. Budući da se ovaj proces izvodi na temelju izvedbe na skupu za provjeru valjanosti, neke informacije o ovom skupu za provjeru valjanosti propuštaju u sam model, tj. preklapaju se sa skupom za provjeru valjanosti, iako se model nikada na njemu izravno ne trenira za parametre koji se mogu naučiti. Iz tog razloga, zajamčeno je da će model s fino podešenim hiperparametrima na skupu za provjeru valjanosti dobro funkcionirati na istom skupu za provjeru valjanosti. Stoga je potpuno nevidljiv skup podataka, odnosno zaseban skup testova, neophodan za odgovarajuću evaluaciju performansi modela, jer je konačni cilj izvedba modela na nikada prije viđenim podacima sa visokom točnošću, odnosno generalizacija. [21]

Trening mreže je proces pronalaženja filtera u slojevima konvolucije i težina u potpuno povezanim slojevima koji minimiziraju razlike između predviđanja izlaza i zadanih oznaka temeljne istine na skupu podataka za trening. Algoritam propagacije pogreške unatrag je metoda koja se obično koristi za treniranje neuronskih mreža gdje funkcija gubitka i optimizacijski algoritam gradijenta spusta igraju bitnu ulogu. Izvedba modela pod određenim filterima i težinama izračunava se funkcijom gubitka unaprijednom propagacijom (engl. forward propagation) na skupu podataka za obuku, a parametri koji se mogu naučiti, odnosno filteri i težine, ažuriraju se prema vrijednosti gubitka putem backpropagation algoritma i gradijentnog spusta. [21]

## 4.5. Funkcija gubitka

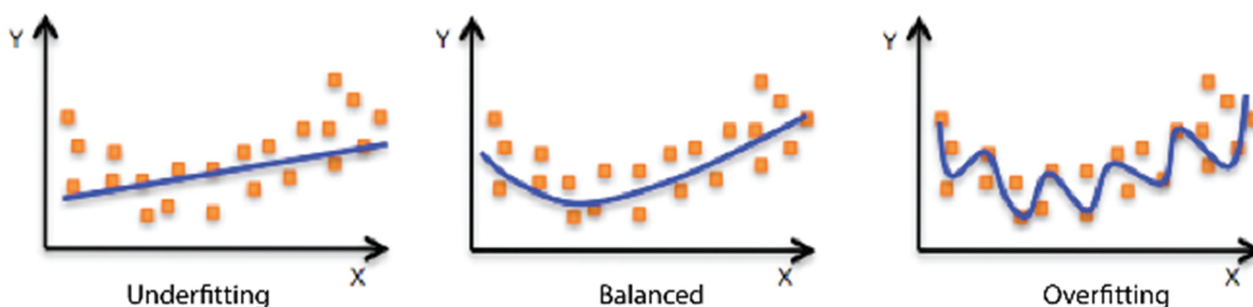
Funkcija gubitka, koja se također naziva i funkcija troškova, mjeri kompatibilnost između predviđanja izlaza mreže putem unaprijedne propagacije i zadanih oznaka temeljne istine. Obično korištena funkcija gubitka za višeklasnu klasifikaciju je unakrsna entropija, dok se srednja kvadratna pogreška obično primjenjuje na regresiju na kontinuirane vrijednosti. Vrsta funkcije gubitka jedan je od hiperparametara i treba je odrediti prema zadanim zadacima. [21]

## 4.6. Prenaučenost i podučenost

Prenaučenost (engl. overfitting) se odnosi na situaciju u kojoj model uči statističke pravilnosti specifične za skup za treniranje, tj. završi memoriranje irelevantnog šuma umjesto učenja signala i, stoga, slabije radi na sljedećem novom skupu podataka. Ovo je jedan od glavnih izazova u strojnom učenju, budući da se prenaučeni model ne može generalizirati na nikad viđene podatke. U tom smislu, skup testova igra ključnu ulogu u pravilnoj evaluaciji izvedbe modela strojnog učenja. Rutinska provjera za prepoznavanje prekomjernog prilagođavanja podacima treninga je praćenje gubitka i točnosti skupova za treniranje i validaciju. [21]

Ako model ostvaruje visoke rezultate u fazi treniranja, a niske na skupu za provjeru valjanosti, smatra se da model ima visoku varijancu te da je model prenaučen. Visoka prenaučenost se rješava prikupljanjem više podataka, regularizacijom ili promjenom arhitekture mreže.

Model također može biti podučen (engl. underfitting), ili imati visoku pristranost (engl. bias). Načini kako bi se riješio problem visoke podučenosti su povećavanje mreže, dakle veći broj hiperparametara, trenirati duže ili promijeniti arhitekturu mreže. [27]



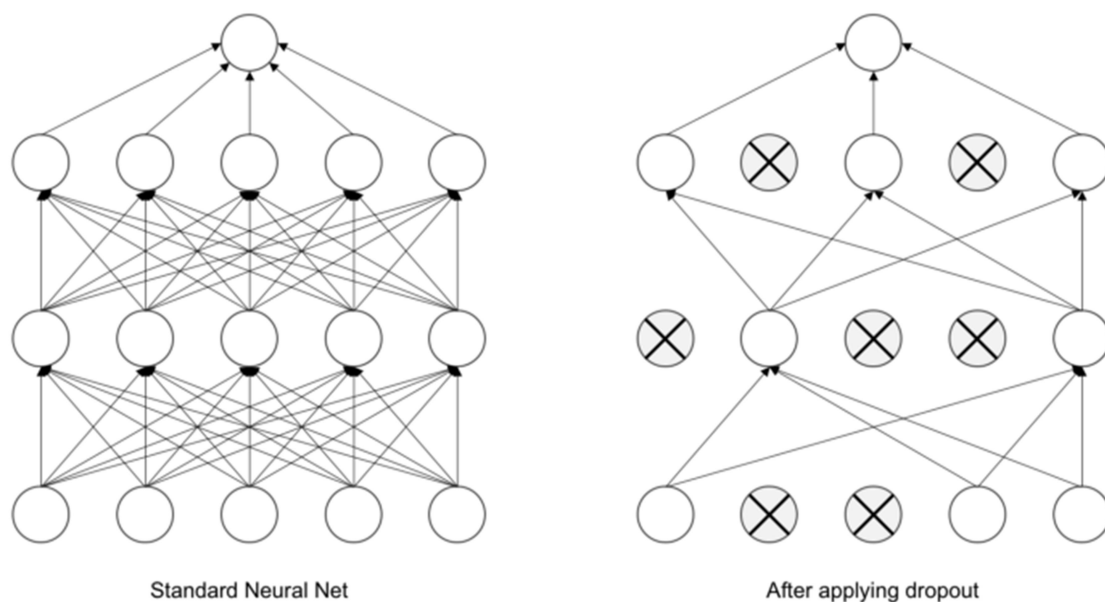
Slika 13: grafovi podučenosti, balansa i prenaučenosti

## 4.7. Regularizacija

Predloženo je nekoliko metoda za minimiziranje prenaučivosti. Najbolje rješenje za smanjenje prenaučivosti je dobivanje više podataka o treningu. Model obučen na većem skupu podataka obično bolje generalizira, iako to nije uvijek moguće postići u medicinskom snimanju. [21]

### 4.7.1. Ispadanje neurona

Ispadanje neurona (engl. dropout) je nedavno uvedena tehnika regularizacije gdje se nasumično odabrane aktivacije postavljaju na 0 tijekom treninga, tako da model postaje manje osjetljiv na specifične težine u mreži. [21]



Slika 14: Neuronska mreža prije i nakon nestajanja neurona

## 4.7.2. Rano zaustavljanje

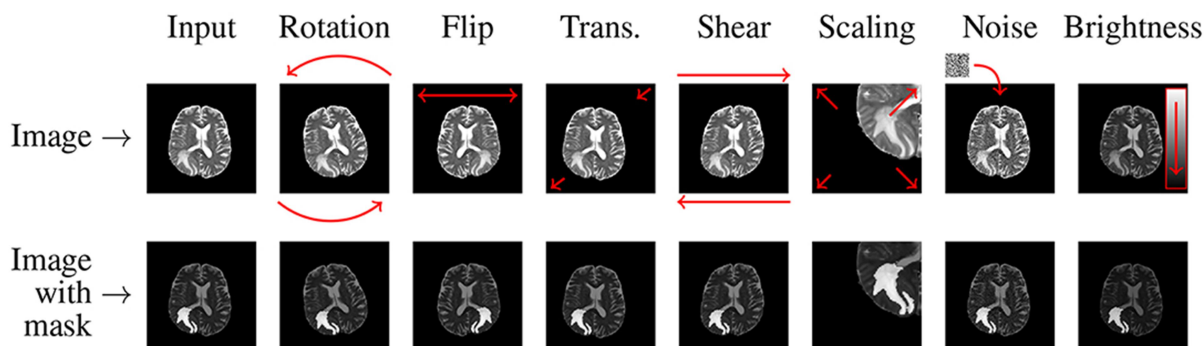
Rano zaustavljanje (engl. early stopping) je metoda regularizacije kod koje se nakon svake epohe vrši validacija. Nakon određenih epoha, kada se ustvrdi da se učenje pogoršava zaustavlja se treniranje. [28]



Slika 15: ilustracija ranog zaustavljanja

## 4.7.3. Augumentacija podataka

Augumentacija podataka je također učinkovita za smanjenje prenaučnosti, što je proces modificiranja podataka o obuci putem slučajnih transformacija, kao što su okretanje, translacije, izrezivanje, zrcaljenje, rotiranje i nasumično brisanje, tako da model neće vidjeti potpuno iste ulaze tijekom iteracije obuke. [21]



Slika 16: primjeri augumentacije slike

#### 4.7.4. Normalizacija grupe

Normalizacijom grupe se prikuplja statistika, prosjek i standardna devijacija značajki, te potom normalizira značajke tako da značajke, odnosno vrijednosti slike, budu s prosjekom nula i standardnom devijacijom jedan. Normalizacijom grupe izbjegava se dominantnost jedne značajke nad drugom značajkom i da se određeni sloj ne određuje uvelike promjenom vrijednosti nekog drugog sloja. Osim toga, treniranje se obavlja brže. [29]

Unatoč tim naporima, još uvijek postoji zabrinutost zbog prenaučivosti skupa za provjeru valjanosti, a ne skupa za trening zbog curenja informacija tijekom finog podešavanja hiperparametara i procesa odabira modela. Stoga je izvješćivanje o izvedbi konačnog modela na zasebnom (nevidljivom) skupu testova, a idealno na vanjskim skupovima podataka za provjeru valjanosti, ako je primjenjivo, ključno za provjeru generalizacije modela. [21]

## 4.8. Optimizacijski algoritmi

U digitalnim slikama, vrijednosti piksela pohranjene su u dvodimenzionalnoj (2D) mreži, tj. u nizu brojeva, a mala mreža parametara nazvana filter, optimizacijski ekstraktor značajki, primjenjuje se na svaku poziciju slike, budući da se značajka može pojaviti bilo gdje na slici. Kako jedan sloj unosi svoj izlaz u sljedeći sloj, izdvojene značajke mogu hijerarhijski i progresivno postati složenije. Proces optimizacije parametara kao što su filteri naziva se trening, koji se izvodi tako da se minimizira razlika između izlaznih podataka i oznaka temeljne istine putem algoritma propagacije pogreške unatrag (engl. backpropagation) i gradijentnog spusta. [21]

Algoritam učenja neuronske mreže sastoji se od dvije faze: unaprijedne propagacije i unazadne propagacije. U prvoj fazi algoritma na ulaze neuronske mreže dovode se ulazi za koje je poznato rješenje, računaju se izlazi mreže za dane ulaze i određuje se pogreška. U drugoj fazi izračunata greška propagira se unazad i računaju se gradijenti pogreške. Gradijent se koristi kako bi znali kako treba promijeniti težine pojedinog neurona da se greška smanji. Pri računanju gradijenata najprije se određuju greške zadnjeg sloja. Zatim se ukupna greška zadnjeg sloja propagira na predzadnji sloj, te se za predzadnji sloj određuje kako je svaki neuron utjecao na grešku. Zatim se određuju komponente gradijenta i ažurira vrijednost težine u ovisnosti o izračunatom gradijentu. [25]

### 4.8.1. Gradijentni spust

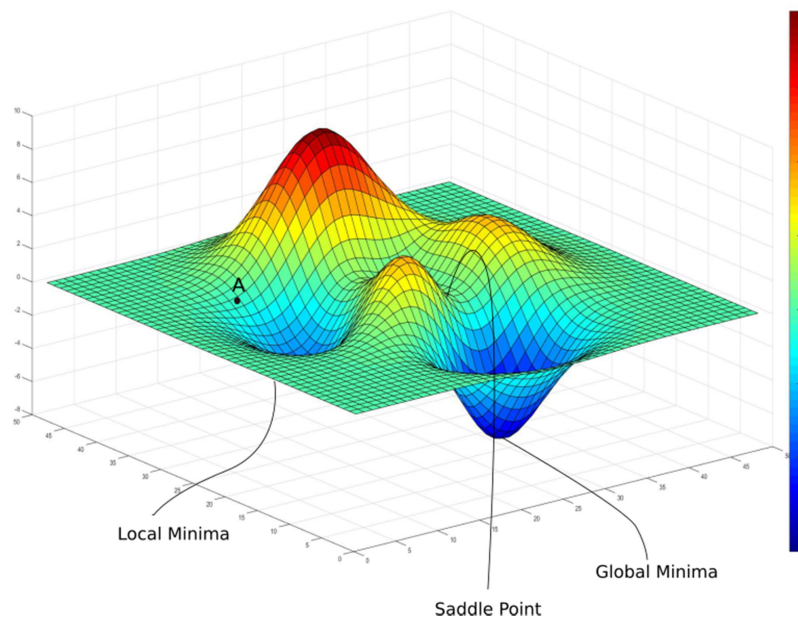
Gradijentni spust (engl. gradient descent) se obično koristi kao optimizacijski algoritam koji iterativno ažurira parametre mreže za učenje, tj. filtere i težine, kako bi se smanjio gubitak. Gradijent funkcije gubitka  $\min f(w)$ , nam daje smjer u kojem funkcija ima najveću stopu povećanja, a svaki parametar koji se može naučiti ažurira se u suprotnom smjeru gradijenta s proizvoljnom veličinom koraka određenom na temelju hiperparametra koji se naziva stopa učenja (engl. learning rate).

Gradijent je djelomična derivacija gubitka u odnosu na svaki parametar koji se može naučiti, a jedno ažuriranje parametra formulira se na sljedeći način:

$$w \leftarrow w - \alpha * \partial L / \partial w$$

gdje  $w$  predstavlja parametar koji se ažurira nakon svake iteracije u gradijentnom spustu krećući se prema globalnom minimumu, točki u kojoj je vrijednost funkcije koju želimo optimizirati manja od vrijednosti danih svim drugim mogućim točkama.  $\alpha$  predstavlja stopu učenja, a  $L$  predstavlja funkciju gubitka.

Važno je napomenuti da je u praksi stopa učenja jedan od najvažnijih hiperparametara koji treba postaviti prije početka treninga. Stopa učenja  $\alpha$  je koeficijent brzine konvergiranja. Ako je  $\alpha$  prevelik, onda dolazi do divergencije, a ako je  $\alpha$  premalen, onda dolazi do spore konvergencije. [21, 24]



Slika 17: prikaz gradijentnog spusta te lokalni i globalni minimum



## 4.8.2. Stohastički gradijentni spust

Stohastički gradijentni spust (SGD) je optimizacijski algoritam koji uzorkuje samo jedan uzorak za trening iz skupa podataka koji se koristi za izračunavanje gradijenta po iteraciji. Budući da SGD nasumično uzorkuje jedan po jedan iz skupa podataka, na kraju mora proći kroz cijeli skup podataka. Jedan prolaz kroz cijeli skup podataka naziva se epoha. Epohe se mogu koristiti u usporedbama između gradijentnog spuštanja i SGD-a za usporedbu brzine kojom se početni parametri  $w$  približavaju svojim optimalnim vrijednostima. [24]

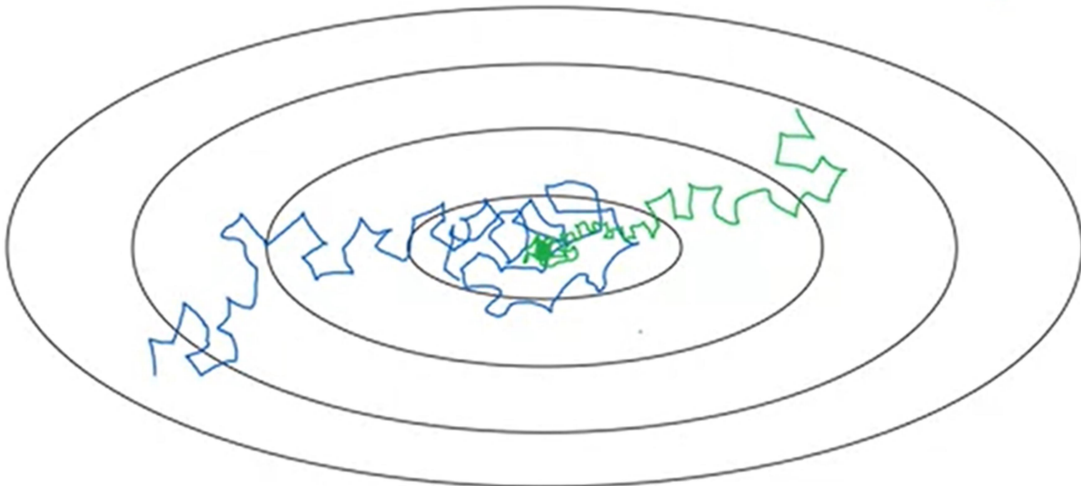
U praksi, iz razloga kao što su ograničenja memorije, gradijenti funkcije gubitka s obzirom na parametre izračunavaju se korištenjem podskupa skupa podataka za treniranje (engl mini-batch) i primjenjuju se na ažuriranja parametara. Ova metoda se zove gradijentni spust manjih grupa (engl mini-batch gradient descent, MBGD).

Osim toga, predložena su i naširoko korištena mnoga poboljšanja algoritma gradijenta spuštanja, kao što su SGD s momentom, RMSprop (engl Root Mean Square Prop) i Adam koji kombinira gradijentni spust s momentom zajedno s RMSprop-om. [21]

### 4.8.3. Opadanje stope učenja

Dok trenira neuronske mreže sa stohastičkim ili mini-batch gradijentom spusta i konstantnom stopom učenja, algoritam obično konvergira prema minimumima na šuman način (manje šumno u MBGD-u) i na kraju oscilira daleko od stvarnih minimuma. Kako bi prevladali ovaj scenarij, istraživači iznose prijedlog opadanja stope učenja (engl. learning rate decay) tijekom vremena što pomaže mreži da konvergira na minimum i izbjegne oscilacije.

Na slici 18. gdje imamo stalnu stopu učenja, koraci koje poduzima algoritam (prikazan plavom linijom) tijekom iteracije prema minimumima su toliko šumni da se nakon određenih iteracija čini da lutaju oko minimuma i zapravo ne konvergiraju. Ali kada se stopa učenja smanjuje tijekom vremena (zeleno linija), budući da je stopa učenja velika, u početku još uvijek imamo relativno brzo učenje, ali kako tendencija ka minimalnoj stopi učenja postaje sve manja i manja, na kraju oscilira u manjem području oko minimuma, a ne luta daleko od njega. Kako vrijeme prolazi, kako algoritam uči kroz razne epohe, tako se koeficijent za učenje s vremenom smanjuje.



Slika 18: konvergiranje algoritma

Formula koja se može koristiti za opadanje stope učenja  $\alpha$  je:

$$\alpha = (1 / (1 + \text{decayRate} \times \text{epochNumber})) * \alpha_0$$

Eksponencijalno smanjenje stope za učenje je definirano formulom:

$$\alpha = (\text{decayRate}^{\text{epochNumber}}) * \alpha_0, \quad \alpha < 1$$

Također se koristi i diskretno smanjenje stope za učenje, odnosno za određene epohe postoji diskretna vrijednost stope za učenje. [30]

## 5. Grad-CAM

Grad-CAM (engl. Gradient-weighted Class Activation Mapping) je metoda za izradu toplinskih mapa primijenjenih na neuronsku mrežu nakon što je njezino osposobljavanje završeno i parametri su fiksirani. Jedan od najvećih problema dubokog učenja je interpretabilnost modela, ključna komponenta u razumijevanju modela i otklanjanju pogrešaka modela. U praksi se modeli dubokog učenja tretiraju kao metode "crne kutije" i mnogo puta nemamo ideju koji se niz neurona aktivirao tijekom zaključivanja ili predviđanja te kako je mreža došla do konačnog rezultata. Koristeći Grad-CAM, može se vizualno potvrditi gdje mreža gleda i provjeriti gleda li ispravne uzorke na slici. Primijenjen je u brojnim istraživačkim područjima, a posebno je popularan u medicinskim slikama.

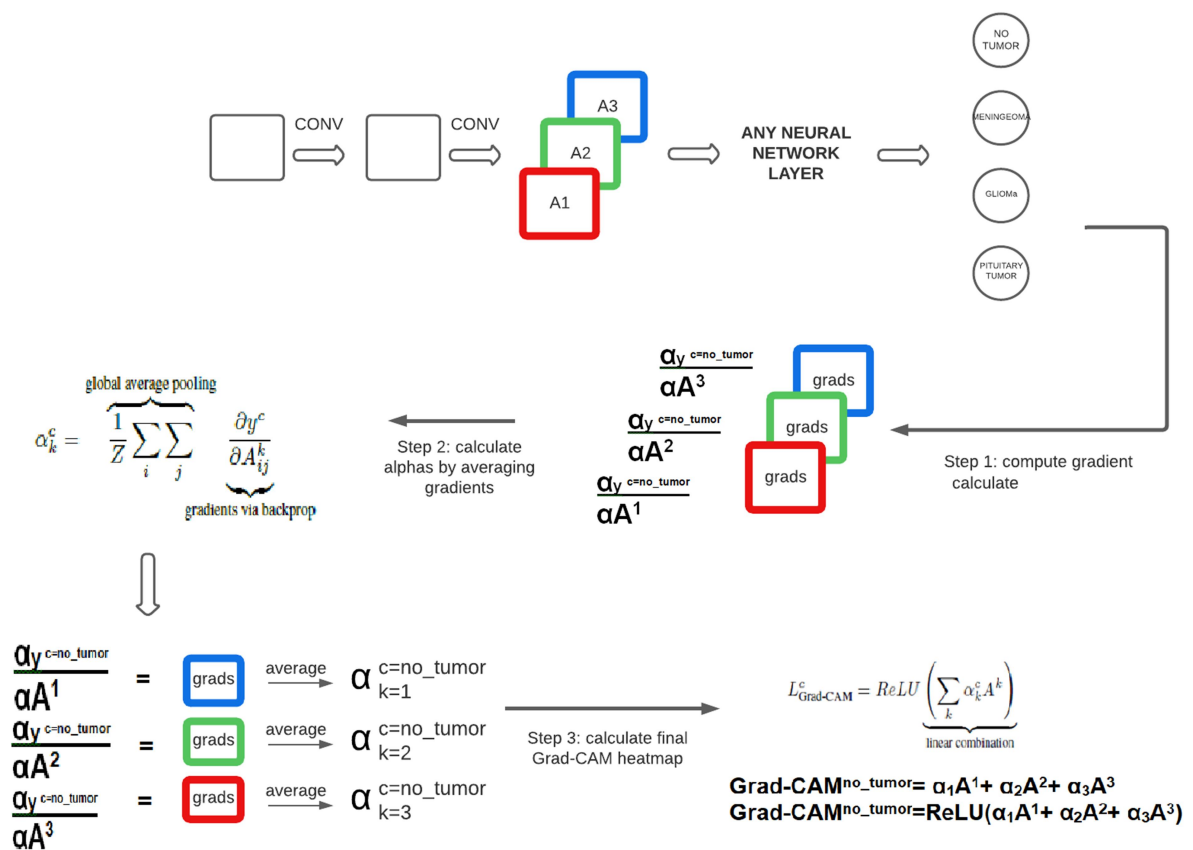
Grad-CAM ne zahtijeva određenu CNN arhitekturu i generalizacija je mapiranja aktivacije klase (CAM), metode koja zahtijeva korištenje specifične arhitekture i primjenu globalnog prosjeka sažimanja (engl. Global Average Pooling, GAP) na konačne konvolucijske mape značajki. Osnovna ideja Grad-CAM-a je iskoristiti prostorne informacije koje su sačuvane kroz konvolucijske slojeve kako bi se razumjelo koji su dijelovi ulazne slike bili ključni za odluku o klasifikaciji. Koristi značajke dobivene iz posljednjeg konvolucijskog sloja CNN-a jer imaju najbolji kompromis između semantike visoke razine i detaljnih prostornih informacija.

Za dobivanje mape lokalizacije diskriminirajuće klase, Grad-CAM izračunava ocjenu gradijenta za danu klasu s obzirom na mape značajki posljednjeg konvolucijskog sloja. Ovi gradijenti su skupljeni u odnosu na globalni prosjek kako bi se dobili ponderi važnosti. Slično CAM-u, Grad-CAM toplinska karta je ponderirana kombinacija mapa značajki praćena ReLU funkcijom.

Specifičan je za klasu, što znači da može proizvesti zasebnu vizualizaciju za svaku klasu prisutnu na slici. Ako postoji deset mogućih izlaznih klasa, tada za određenu ulaznu sliku možete izraditi deset različitih Grad-CAM toplinskih mapa, jednu toplinsku mapu za svaku klasu.

Skica (slika 19.) prikazuje dijelove modela neuronske mreže relevantne za Grad-CAM te korake u izradi toplinske mape. CNN se sastoji od nekoliko konvolucijskih slojeva (prikazano kao "conv" na skici). Mape značajki koje proizvodi završni konvolucijski sloj prikazane su kao A1, A2 i A3. U ovom trenutku, za CAM potrebno je napraviti globalno prosječno udruživanje nakon čega slijedi potpuno povezani sloj. Za Grad-CAM može se učiniti bilo što, kao što je prikazano na skici može biti "bilo koji sloj neuronske mreže". Jedini uvjet je da se slojevi nakon A1, A2 i A3 moraju razlikovati kako bi se mogao dobiti gradijent. Na kraju su rezultati klasifikacije za gliom, meningeom, tumor hipofize te beztumorske snimke.

Razlika između CAM-a i Grad-CAM-a je u načinu na koji se mape značajki A1, A2 i A3 ponderiraju kako bi se napravila konačna toplinska karta. U CAM-u se ove mape značajki ponderiraju koristeći težine preuzete iz posljednjeg potpuno povezanog sloja mreže. U Grad-CAM-u se ponderiraju karte značajki pomoću "alfa vrijednosti" koje se izračunavaju na temelju gradijenata. Stoga Grad-CAM ne zahtijeva posebnu arhitekturu, jer se gradijenti mogu izračunati kroz bilo koji sloj neuronske mreže. "Grad" u Grad-CAM-u označava "gradijent".



Slika 19: princip rada Grad-CAM-a

Recimo da je  $y^c$  neobrađeni izlaz neuronske mreže za klasu c, prije nego što se softmax primijeni za pretvaranje sirovog rezultata u vjerojatnost. Grad-CAM se primjenjuje na neuronsku mrežu koja se obučava. Težine neuronske mreže su fiksne. Ubacuje se slika u mrežu kako bi se izračunala Grad-CAM toplinska mapa za tu sliku za odabranu klasu interesa.

Prvi korak je računanje gradijenta. Posebna vrijednost gradijenta izračunata u prvom koraku ovisi o odabranoj ulaznoj slici, jer ulazna slika određuje mape značajki  $A^k$  kao i konačnu ocjenu klase  $y^c$  koja se proizvodi. Za dvodimenzionalnu ulaznu sliku, ovaj gradijent je trodimenzionalni, istog oblika kao mape značajki. Postoji  $k$  mapa značajki, svaka visine  $v$  i širine  $u$ , tj. zajedno mape značajki imaju oblik  $[k, v, u]$ . To znači da će gradijenti izračunati u prvom koraku također biti oblika  $[k, v, u]$ . Na skici je  $k=3$ , tako da postoje tri mape značajki  $u \times v$  i tri gradijenta  $u \times v$ .

U sljedećem koraku vrši se izračun alfa vrijednosti usrednjavanjem gradijenata. Alfa vrijednost za klasu  $c$  i kartu značajki  $k$  koristit će se kao težina primijenjena na kartu značajki  $A^k$ . S obzirom da gradijenti imaju oblik  $[k, v, u]$ , zbraja se visina  $v$  i širina  $u$ , a dobiveni rezultat je oblika  $[k, 1, 1]$  ili pojednostavljeno, samo  $[k]$ . Ovo su  $k$  alfa vrijednosti.

Na kraju se računa konačna Grad-CAM toplinska mapa. Svaku alfa vrijednost služi kao težina odgovarajuće mape značajki i izračunava se ponderirani zbroj mapa značajki kao konačnu Grad-CAM toplinska mapu. Zatim se primjenjuje ReLU operacija kako bi se naglasile samo pozitivne vrijednosti i pretvorile sve negativne vrijednosti u 0. Izlaz Grad-CAM-a je "klasno diskriminirajuća mapa lokalizacije", tj. toplinska mapa gdje vrući dio odgovara određenoj klasi:

$$L_{GradCAM}^c \in \mathbb{R}^{u \times v}$$

Ako se mreža ne aktivira oko odgovarajućih uzoraka odnosno objekata na slici, mreža nije ispravno naučila temeljne obrasce u određenom skupu podataka te je potrebno je preispitati postupak obuke. Nekad je potrebno prikupiti dodatne podatke.

## 6. Transferno učenje

Obilje dobro označenih podataka u medicinskom snimanju poželjno je, ali rijetko dostupno zbog troškova i potrebnog posla radioloških stručnjaka. Postoji nekoliko dostupnih tehnika za učinkovitu obuku modela na manjem skupu podataka: povećanje podataka i transferno učenje. Povećanje podataka ukratko je obrađeno ranije, stoga se ovaj odjeljak fokusira na transferno učenje.

Transferno učenje uobičajena je i učinkovita strategija za treniranje mreže na malom skupu podataka, gdje je mreža unaprijed obučena na iznimno velikom skupu podataka, kao što je ImageNet, koji sadrži 1,4 milijuna slika s 1000 klasa, zatim se ponovno koristi i primjenjuje na zadani zadatak interesa. Temeljna pretpostavka transfernog učenja je da se generičke značajke naučene na dovoljno velikom skupu podataka mogu dijeliti među naizgled različitim skupovima podataka. Ova prenosivost naučenih generičkih značajki jedinstvena je prednost dubokog učenja koje se čini korisnim u raznim zadacima domene s malim skupovima podataka. Trenutno su mnogi modeli unaprijed obučeni za skup podataka izazova ImageNet otvoreni za javnost i lako dostupni, zajedno s njihovim naučenim jezgrama i težinama. [21]

Jedan nedostatak transfernog učenja su njegova ograničenja na ulazne dimenzije. Ulazna slika mora biti dvodimenzionalna s tri kanala relevantna za RGB jer se ImageNet skup podataka sastoji od dvodimenzionalnih slika u boji koje imaju tri kanala (RGB: crvena, zelena i plava), dok medicinske slike u sivim tonovima imaju samo jedan kanal (razine sive). S druge strane, visina i širina ulazne slike mogu biti proizvoljne, ali ne premale. [21]

Povijest dizajna konvolucijskih neuronskih mreža započela je s modelima u stilu LeNeta, koji su bili jednostavni nizovi konvolucija za ekstrakciju značajki i operacije maksimalnog udruživanja za prostorno poduzorkovanje. Godine 2012. te su ideje dorađene u AlexNet arhitekturu, gdje su se operacije konvolucije ponavljale više puta između operacija max-poolinga, omogućujući mreži da nauči bogatije značajke na

svakoj prostornoj razini. Ono što je uslijedilo bio je trend da se ovaj stil mreže učini sve dubljim, uglavnom potaknut godišnjim natjecanjem ILSVRC; prvo sa Zeilerom i Fergusom 2013., potom iduće godine s VGG arhitekturom. Zatim se pojavio novi stil mreže, Inception arhitektura, koju su predstavili Szegedy i tim 2014. Kao GoogLeNet (Inception V1), kasnije dorađen kao Inception V2, Inception V3 i nedavno Inception-ResNet. [21, 38]

U ovom radu, cilj je proučiti učinkovitost modela koji imaju najvišu najsuvremeniju točnost i najbolje karakteristike. U tablici ispod ova četiri CNN modela odabrana su prema navedenim karakteristikama na stranici Kerasa, odnosno na temelju veličine mreže u megabajtima, točnosti prema njihovim top-1 i top-5, broju parametara koji se mogu trenirati, dubini, te brzini računanja na CPU i GPU.

| Model          | Size (MB) | Top-1 Accuracy | Top-5 Accuracy | Parameters | Depth | Time (ms) per inference step (CPU) | Time (ms) per inference step (GPU) |
|----------------|-----------|----------------|----------------|------------|-------|------------------------------------|------------------------------------|
| Xception       | 88        | 79.0%          | 94.5%          | 22.9M      | 81    | 109.4                              | 8.1                                |
| ResNet50V2     | 98        | 76.0%          | 93.0%          | 25.6M      | 103   | 45.6                               | 4.4                                |
| InceptionV3    | 92        | 77.9%          | 93.7%          | 23.9M      | 189   | 42.2                               | 6.9                                |
| EfficientNetB0 | 29        | 77.1%          | 93.3%          | 5.3M       | 132   | 46.0                               | 4.9                                |

Slika 20: karakteristike mrežnih arhitektura Xception, ResNet50v2, Inceptionv3 i EfficientNetb0

Xception i Inceptionv3, oba imaju po oko 23M parametara, te postoje manje razlike točnosti njihovih top-5 i top-1. Xception ima manje megabajta i dubinu, ali zahtjeva puno više računanja, što znači više potrebnog vremena i energije za obuku.

ResNet50v2 ima nešto veći broj parametara u usporedbi sa prethodna dva, te nešto manju točnost. Što se tiče ostalih karakteristika (veličine, dubine i količine računanja) nalazi se između Xceptiona i Inceptiona.

EfficientNetB0 ima najmanju veličinu (3 puta manju od ostalih), bolju točnost, manju dubinu i malo veće vrijeme računanja na CPU i GPU od ResNet50v2.

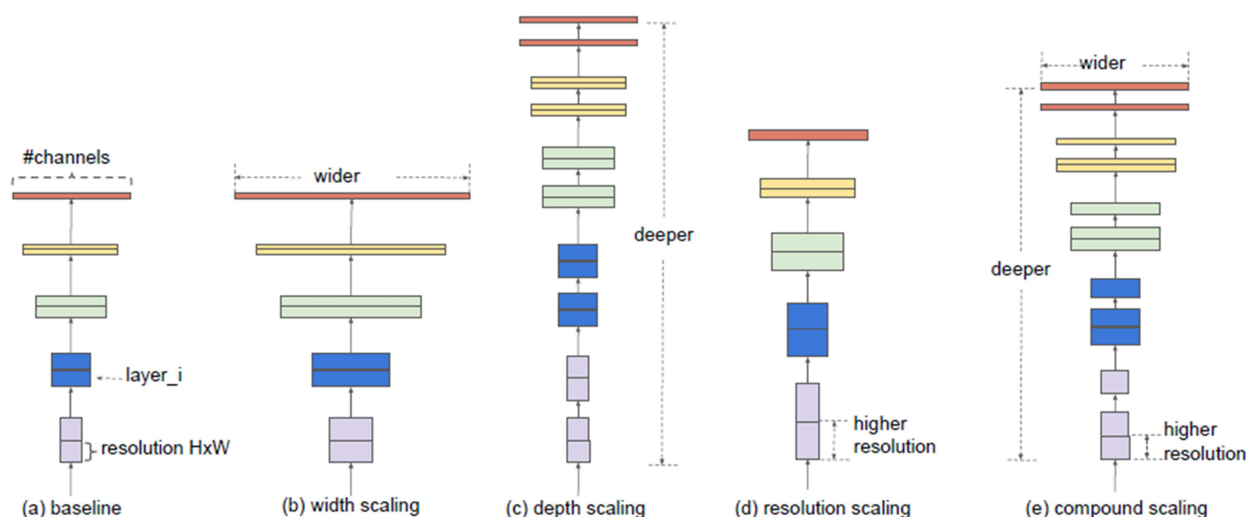


## 6.1. Arhitekture mreža

### 6.1.1. EfficientNet

Općenito, modeli su napravljeni preširoki, duboki ili s vrlo visokom rezolucijom. Povećanje ovih karakteristika u početku pomaže modelu, ali se brzo zasiti i napravljeni model samo ima više parametara i stoga nije učinkovit.

Skaliranje bilo koje dimenzije širine, dubine ili razlučivosti mreže poboljšava točnost, ali dobitak točnosti smanjuje se za veće modele. Za slike veće rezolucije trebalo bi povećati dubinu mreže, tako da veća receptivna polja mogu pomoći u hvatanju sličnih značajki koje uključuju više piksela u većim slikama. Sukladno tome, također bi trebalo povećati širinu mreže kada je razlučivost veća, kako bi uhvatili sitnije uzorke s više piksela na slikama visoke razlučivosti. Ove intuicije sugeriraju da je, umjesto konvencionalnog jednodimenzionalnog skaliranja, potrebno koordinirati i uravnotežiti različite dimenzije skaliranja. [33]



Slika 21: Skaliranje modela. (a) primjer je osnovne mreže; (b)-(d) su konvencionalno skaliranje koje povećava samo jednu dimenziju širine, dubine ili razlučivosti mreže; (e) metoda skaliranja koja ravnomjerno skalira sve tri dimenzije s fiksnim omjerom

U EfficientNet-u oni su skalirani na principijelniji način, tj. postupno se sve povećava. Sa znatno manjim brojem parametara, ovi modeli su učinkoviti i daju bolje rezultate.

| Stage<br>$i$ | Operator<br>$\hat{\mathcal{F}}_i$ | Resolution<br>$\hat{H}_i \times \hat{W}_i$ | #Channels<br>$\hat{C}_i$ | #Layers<br>$\hat{L}_i$ |
|--------------|-----------------------------------|--|--------------------------|------------------------|
| 1            | Conv3x3                           | $224 \times 224$                           | 32                       | 1                      |
| 2            | MBCConv1, k3x3                    | $112 \times 112$                           | 16                       | 1                      |
| 3            | MBCConv6, k3x3                    | $112 \times 112$                           | 24                       | 2                      |
| 4            | MBCConv6, k5x5                    | $56 \times 56$                             | 40                       | 2                      |
| 5            | MBCConv6, k3x3                    | $28 \times 28$                             | 80                       | 3                      |
| 6            | MBCConv6, k5x5                    | $14 \times 14$                             | 112                      | 3                      |
| 7            | MBCConv6, k5x5                    | $14 \times 14$                             | 192                      | 4                      |
| 8            | MBCConv6, k3x3                    | $7 \times 7$                               | 320                      | 1                      |
| 9            | Conv1x1 & Pooling & FC            | $7 \times 7$                               | 1280                     | 1                      |

Slika 22: arhitektura EfficientNetB0 mreže

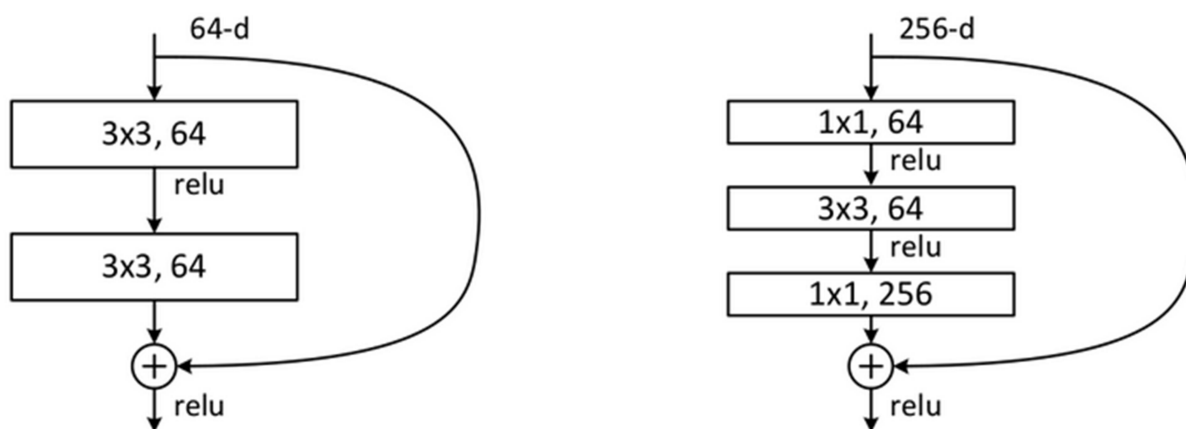
Prva stvar svake mreže je njezina osnova nakon koje kreće svo eksperimentiranje s arhitekturom koje je uobičajeno u svih osam modela i završnih slojeva. Nakon toga svaki od njih sadrži 7 blokova. Ovi blokovi nadalje imaju različit broj podblokova čiji se broj povećava kako prelazimo s EfficientNetB0 na EfficientNetB7. Ukupan broj slojeva u EfficientNetB0 je 237, a u EfficientNet-B7 iznosi 813. Lako je vidjeti razliku među svim modelima i postupno su povećavali broj podblokova. Tablica gore prikazana označava veličinu kernela za operacije konvolucije zajedno s rezolucijom, kanalima i slojevima u EfficientNet-B0.

U testovima je EfficientNets pokazao i veću točnost i bolju učinkovitost u odnosu na postojeće CNN-ove, smanjujući veličinu parametra i FLOPS-a (engl. floating-point operations per second) za red veličine. Jedan od modela, EfficientNetB7, koji je 8,4 puta manji i 6,1 puta brži od CNN Gpipe visokih performansi, dosegao je 84,4% odnosno 97,1% top-1 i top-5 točnosti na ImageNetu. A u usporedbi s popularnim ResNet-50, drugi EfficientNet, EfficientNetB4, koristio je slične FLOPS-ove dok je poboljšavao top-1 točnost sa ResNet-50 76,3% na 82,6%. [33]

## 6.1.2. Resnet

Duboke rezidualne mreže, ResNet, kao osnovni građevni blok koriste rezidualne blokove za poboljšanje točnosti modela. Koncept "preskakanja veza" (engl. skip connections), koji leži u srži rezidualnih blokova, snaga je ove vrste neuronske mreže.

Ove veze za preskakanje rade na dva načina. Prvo, ublažavaju problem nestajanja gradijenta postavljanjem alternativnog prečaca kroz koji gradijent prolazi. Osim toga, rezidualni blokovi znatno olakšavaju slojevima učenje funkcija identiteta. To osigurava da viši slojevi modela ne rade lošije od nižih slojeva. Kao rezultat toga, ResNet poboljšava učinkovitost dubokih neuronskih mreža s više neuronskih slojeva, a minimizira postotak pogrešaka. Drugim riječima, veze za preskakanje dodaju izlaze iz prethodnih slojeva na izlaze naslaganih slojeva, što omogućuje treniranje mnogo dubljih mreža nego što je prije bilo moguće. [34, 35]



Slika 23: koncept preskakanja veza

Umjesto iz  $x \rightarrow F(x)$ , mreža uči preslikavanje iz  $x \rightarrow F(x)+G(x)$ . Kada je dimenzija ulaza  $x$  i izlaza  $F(x)$  jednaka, funkcija  $G(x) = x$  je funkcija identiteta, a veza prečaca naziva se identitetska veza. Identično mapiranje se uči nuliranjem težina u srednjem sloju tijekom treninga budući da je lakše nulirati težine nego ih staviti na jedan.

U slučaju kada se dimenzije  $F(x)$  razlikuju od  $x$  (zbog duljine koraka  $> 1$  u konvolucijskim slojevima između), implementira se veza projekcije umjesto veze identiteta. Funkcija  $G(x)$  mijenja dimenziju ulaza  $x$  u dimenziju izlaza  $F(x)$ . U izvornom radu razmatrane su dvije vrste preslikavanja.

- Mapiranje koje se ne može trenirati (padding): ulaz  $x$  jednostavno je dopunjen nulama kako bi se dimenzija poklopila s onom  $F(x)$ .
- Mapiranje koje se može trenirati (conv layer):  $1 \times 1$  konvolucijski sloj se koristi za mapiranje  $x$  u  $G(x)$ . U cijeloj mreži se prostorne dimenzije ili zadržavaju iste ili prepolovljene, a dubina ili ostaje ista ili se udvostručuje, a umnožak širine i dubine nakon svakog konvolucijskog sloja ostaje isti, tj. 3584.  $1 \times 1$  konvolucijski slojevi koriste se za polovicu prostorne dimenzije i udvostručenje dubine korištenjem duljine koraka od 2 i višestruke od takvih filtera. Broj  $1 \times 1$  konv slojeva jednak je dubini  $F(x)$ . [34]

| layer name | output size      | 18-layer  | 34-layer  | 50-layer  | 101-layer  | 152-layer  |
|------------|------------------|---|---|---|--|--|
| conv1      | $112 \times 112$ | 7×7, 64, stride 2   |   |   |  |  |
| conv2_x    | $56 \times 56$   | 3×3 max pool, stride 2  |   |   |  |  |
|            |                  | $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$   | $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$   | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$    | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$     | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$     |
| conv3_x    | $28 \times 28$   | $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$ | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$  | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$   | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$   |
| conv4_x    | $14 \times 14$   | $\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$ | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$ |
| conv5_x    | $7 \times 7$     | $\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$ | $\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$  | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$  |
|            | $1 \times 1$     | average pool, 1000-d fc, softmax  |   |   |  |  |
| FLOPs      |                  | $1.8 \times 10^9$   | $3.6 \times 10^9$   | $3.8 \times 10^9$   | $7.6 \times 10^9$  | $11.3 \times 10^9$   |

Slika 24: arhitekture ResNet modela

Postoji više verzija ResNetXX arhitekture koje rade na istom konceptu gdje 'XX' označava broj slojeva. Najčešće korišteni su ResNet50 i ResNet101. Dakle, Resnet50 se koristi za označavanje varijante koja može raditi s 50 slojeva neuronske mreže. Budući da je problem nestajanja gradijenta riješen, CNN je počeo ulaziti sve dublje i dublje. [34]

ResNet50 je varijanta ResNet modela koji ima 48 konvolucijskih slojeva. Kao što je prikazano u na slici 24, arhitektura Resnet50 započinje konvolucijom s veličinom kernela 7x7 i 64 različite jezgre sve s korakom veličine 2 što nam daje 1 sloj. Slijedi max pool također s veličinom koraka od 2, pa niz konvolucijskih slojeva. U cijeloj mreži koriste se samo dva sloja za spajanje, spomenuti max-pool na početku, a drugi, average pool, na kraju mreže. Završava potpuno povezanim slojem koji sadrži 1000 čvorova i na kraju softmax funkcijom tako da to daje 1 sloj. [36]

Građevni blok ovog modela je modificiran u dizajn uskog grla kako bi osigurali poboljšanu točnost i kraće vrijeme obuke. Veze prečaca ovog modela preskaču troslojni blok, dok su raniji ResNet modeli preskakali dva sloja. Ovo ima mnogo veću točnost od 34-slojnog ResNet modela. 50-slojni ResNet postiže učinak od 3,8 milijardi FLOPS-a. [35]

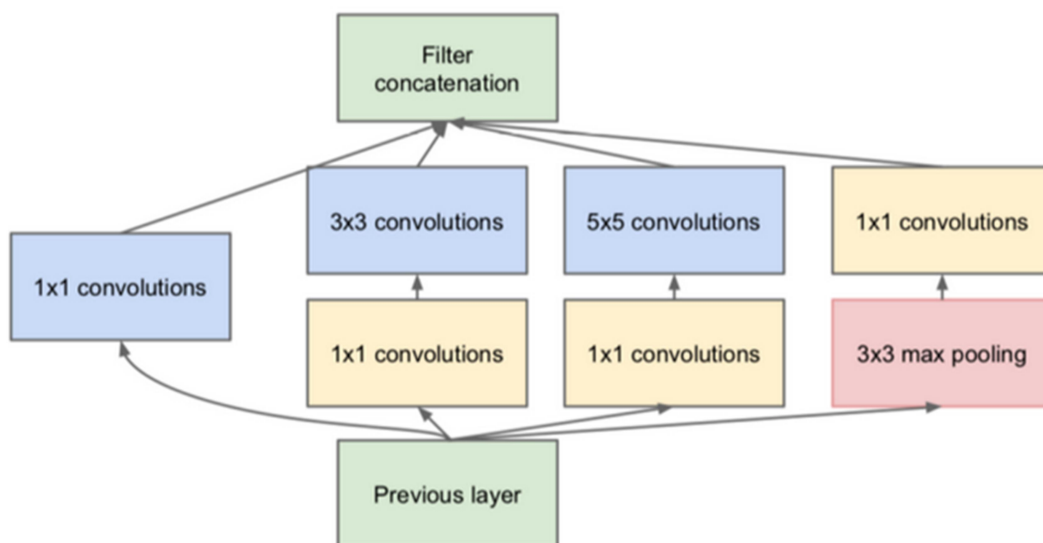
### 6.1.3 Inception

Izvorna verzija ove arhitekture nazvana je GoogLeNet, ali kasnije manifestacije jednostavno su nazvane InceptionvN gdje se N odnosi na broj verzije koji je objavio Google. Glavno obilježje Inception arhitekture je poboljšana iskorištenost računalnih resursa unutar mreže.

U zadatku klasifikacije slika, veličina istaknute značajke može značajno varirati unutar okvira slike. Stoga je odlučivanje o fiksnoj veličini kernela prilično teško. Veći kerneli preferiraju se za globalnije značajke koje su raspoređene na velikom području slike, s druge strane, manji kerneli daju dobre rezultate u otkrivanju značajki specifičnih za područje koje su raspoređene po okviru slike. Za učinkovito prepoznavanje takve značajke promjenjive veličine potrebni su nam kerneli različitih veličina. Cilj Inception modula je djelovati tako da izvlači značajke na više razina izračunavanjem  $1 \times 1$ ,  $3 \times 3$  i  $5 \times 5$  konvolucija unutar istog modula mreže — izlaz ovih filtara se zatim slaže duž dimenzije kanala i prije nego što se unese u sljedeći sloj u mreži. [37]

Inception slojevi se ponavljaju mnogo puta, što dovodi do dubokog modela od 22 sloja (kada se računaju samo slojevi s parametrima ili 27 slojeva ako računamo i slojeve sažimanja) u slučaju GoogLeNet modela. Ukupan broj slojeva (neovisnih građevnih blokova) koji se koriste za izgradnju mreže je oko 100. Međutim, taj broj ovisi o sustavu infrastrukture strojnog učenja koji se koristi. Veličina receptivnog polja u mreži je 224x224 uzimajući RGB kanale boja uz srednje odzimanje. Kada se “mreža u mreži” pristup primijeni na konvolucijske slojeve, metoda se može promatrati kao dodatnih 1x1 konvolucijskih slojeva nakon kojih obično slijedi ispravljena linearna aktivacija. To mu omogućuje jednostavnu integraciju u trenutne kanale CNN-a. 1x1 konvolucije služe kao moduli za smanjenje dimenzija za uklanjanje računalnih uskih grla, koja bi inače ograničila veličinu mreža. To omogućuje ne samo povećanje dubine, već i širine mreža bez značajnog gubitka performansi. [37]

Početa mrežna arhitektura sastoji se od nekoliko početnih modula sljedeće strukture. Svaki početni modul sastoji se od četiri paralelne operacije: 1x1, 3x3 i 5x5 konvolucijskog sloja te sloja maksimalnog sažimanja. 1x1 konvolucijski blokovi (prikazani žutom bojom na slici) koriste se za smanjenje dubine. Rezultati iz četiri paralelne operacije se zatim spajaju po dubini kako bi se formirao blok konkatencije filtera (engl. Filter Concatenation) zelene boje.



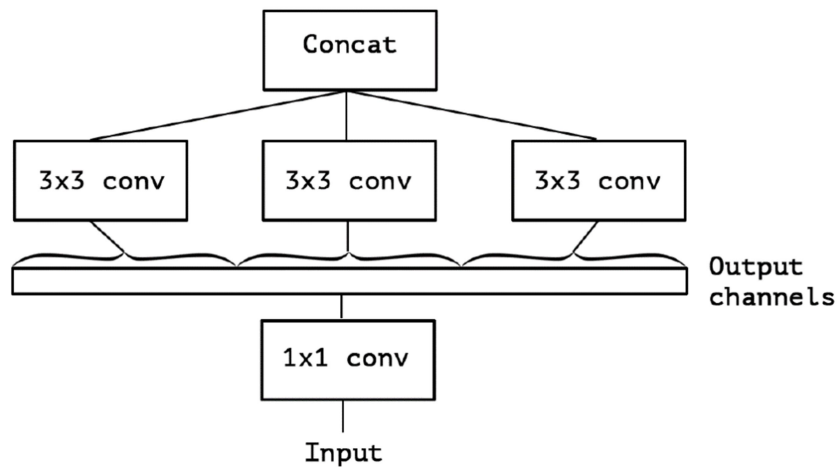
Slika 25: arhitektura Inception mreže

Inception povećava mrežni prostor iz kojeg će se odabrati najbolja mreža putem treniranja. Svaki početni modul može obuhvatiti istaknute značajke na različitim razinama. Globalne značajke uhvaćene su konvolucijskim slojem 5x5, dok je sloj 3x3 sklon hvatanju distribuiranih značajki. Operacija maksimalnog sažimanja odgovorna je za hvatanje značajki niske razine koje se ističu u susjedstvu. Na danoj razini, sve ove značajke se izdvajaju i spajaju prije nego što se unesu u sljedeći sloj. Prepuštamo mreži/obuci da odluči koje značajke imaju najviše vrijednosti i težine u skladu s tim. Recimo ako su slike u skupu podataka bogate globalnim značajkama bez previše značajki niske razine, tada će obučena početna mreža imati vrlo male težine koje odgovaraju 3x3 konvolucijskom kernelu u usporedbi s 5x5 konvolucijskim kernelom. [37]

#### **6.1.4. Xception**

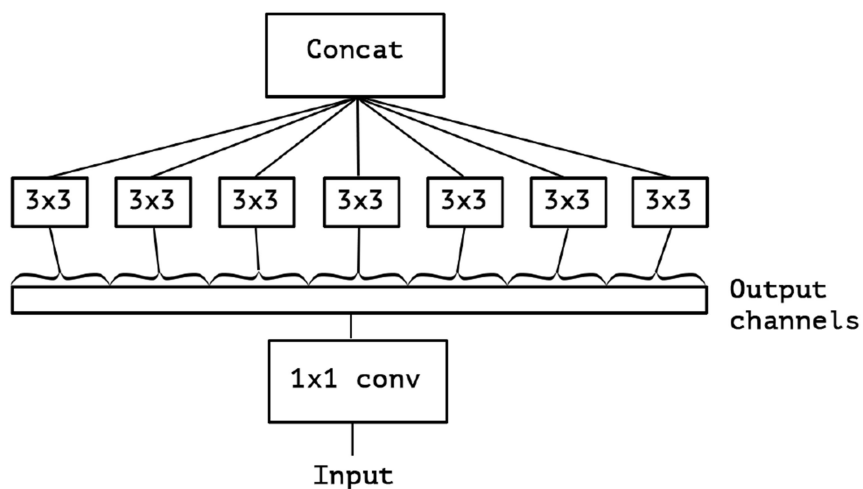
Arhitektura duboke konvolucijske neuronske mreže inspirirana Inceptionom, gdje su Inception moduli zamijenjeni dubinski odvojivim konvolucijama. Mapiranje međukanalnih korelacija i prostornih korelacija u mapama značajki konvolucijskih neuronskih mreža može se u potpunosti razdvojiti. Ova arhitektura, nazvana Xception malo nadmašuje InceptionV3 na skupu podataka ImageNet (za koji je Inception V3 dizajniran) i značajno nadmašuje Inception V3 na većem skupu podataka o klasifikaciji slika koji se sastoji od 350 milijuna slika i 17 000 klasa. Budući da arhitektura Xception ima isti broj parametara kao InceptionV3, poboljšanja performansi nisu posljedica povećanog kapaciteta, već učinkovitijeg korištenja parametara modela. [38]

Izvorna dubinski odvojiva konvolucija sastoji se od konvolucije po dubini, tj. prostorne konvolucije koja se izvodi neovisno nad svakim kanalom ulaza, nakon čega slijedi točkasta konvolucija, tj. konvolucija 1x1, projicirajući izlaz kanala pomoću konvolucije po dubini na novi prostor kanala. Slika 26. prikazuje Inception modul preformuliran kao velika 1x1 konvolucija praćena prostornim konvolucijama koje bi djelovale na segmentima izlaznih kanala koji se ne preklapaju.



Slika 26: strogo ekvivalentna reformulacija pojednostavljenog Inception modula

Modificirana odnosno "ekstremna" verzija Inception modula, prvo bi koristila konvoluciju 1x1 za mapiranje međukanalnih korelacija, a zatim bi zasebno mapirala prostorne korelacije svakog izlaznog kanala. Ovo je prikazano na slici 27.

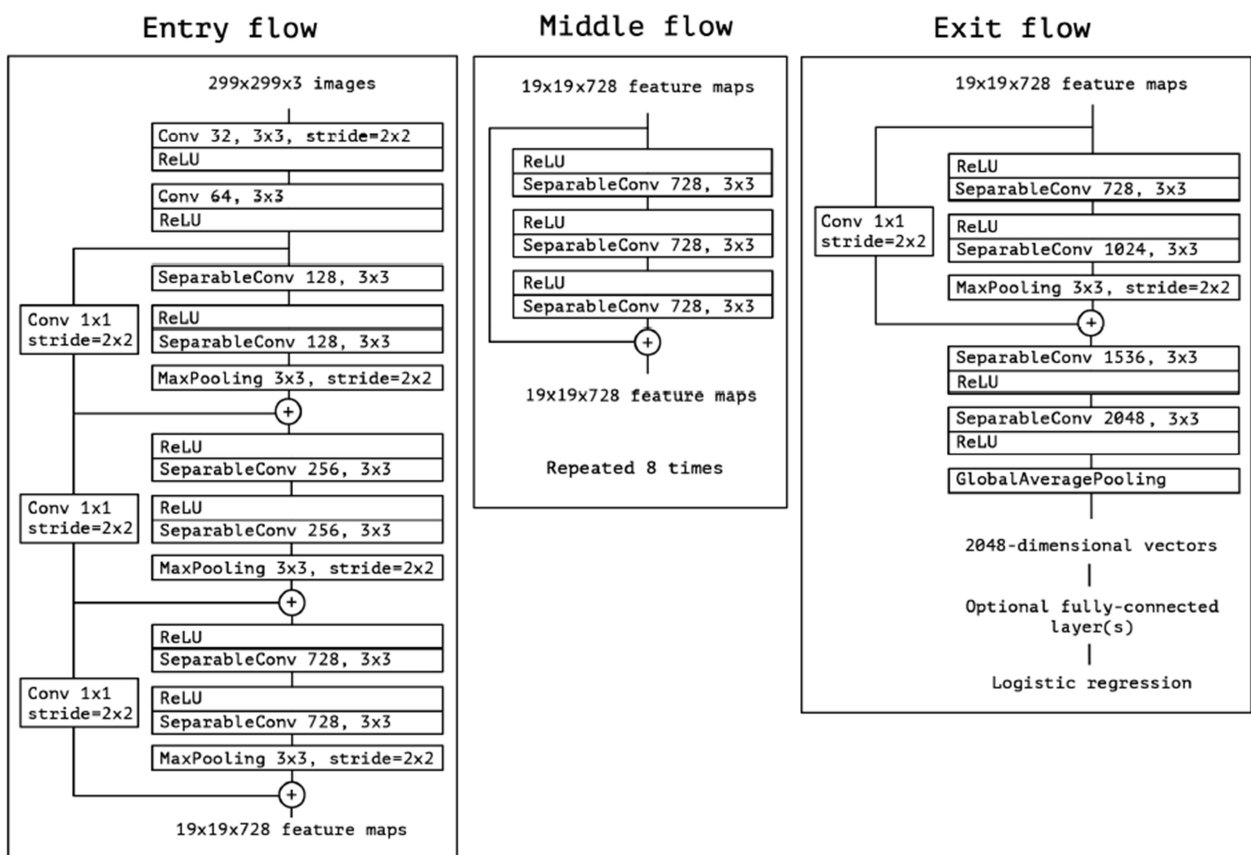


Slika 27: "ekstremna" verzija Inception modula

U izvornom Inception modulu, obje operacije su praćene ReLU nelinearnošću, međutim u Xceptionu, dubinski odvojive konvolucije obično se implementiraju bez nelinearnosti.



Arhitektura Xception ima 36 konvolucijskih slojeva koji tvore bazu za ekstrakciju značajki mreže. Konvolucijsku bazu prati sloj logističke regresije, a po izboru se mogu umetnuti potpuno povezani slojevi prije sloja logističke regresije. 36 konvolucijskih slojeva strukturirano je u 14 modula, od kojih svi imaju linearne rezidualne veze oko sebe, osim prvog i posljednjeg modula. Ukratko, Xception arhitektura je linearni skup slojeva konvolucije koji se mogu odvojiti po dubini s rezidualnim vezama. Potpuni opis specifikacija mreže dan je na slici 28. [38]



Slika 28: Xception arhitektura

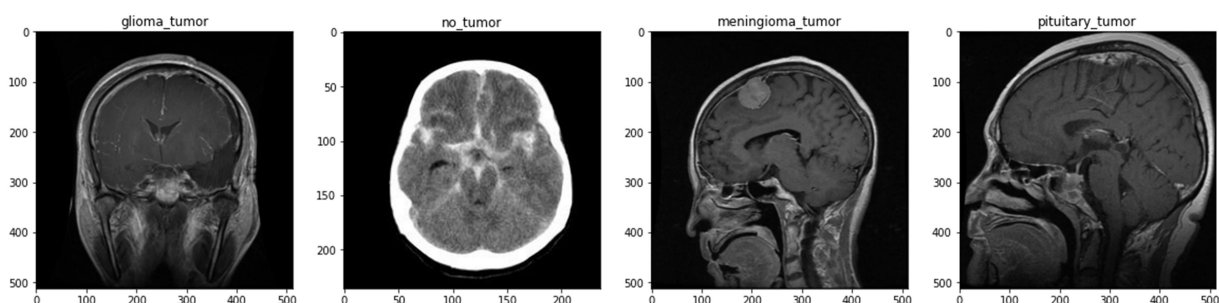
## 7. Metode rada

Za izradu programskog koda korišten je Python programski jezik uz biblioteke Tensorflow, Seaborn i Keras. S ciljem izrade što preciznijeg modela visoke točnosti testirane su 4 poznate arhitekture konvolucijske neuronske mreže za izvedbu klasifikacije MR snimki mozga. S obzirom na izvođenje programa u okruženju Google Colaboratory bilo je potrebno spremati odgovarajuće slike koje čine podatkovni skup na Google disk, a kod se nalazi na poveznici: [https://github.com/luksaana/Brain-tumor-detection/blob/main/Brain\\_tumor\\_detection\\_on\\_MR\\_images\\_using\\_CNN.ipynb](https://github.com/luksaana/Brain-tumor-detection/blob/main/Brain_tumor_detection_on_MR_images_using_CNN.ipynb).

### 7.1. Skup podataka i predobrada

Skup podataka magnetske rezonance (MRI) mozga preuzet je sa stranice Kaggle.com. Sadrži 3264 dvodimenzionalnih slika mozga pacijenata u aksijalnim, koronarnim ili sagitalnim presjecima. Skupovi podataka sadržavali su samo snimku područja interesa i oznaku bolesti. Matrice slike su  $512 \times 512$ .

Mapa podataka podijeljena je na skup za treniranje i skup za testiranje. Te mape sadrže još četiri podmape odnosno klase: slike bez tumora te slike s gliomom, meningiomom i tumorom hipofize. Slika 29. prikazuje primjer svake klase.



Slika 29: primjer slike iz svake klase

Za početak je potrebno napraviti analizu podataka. Važno je analizirati koliko je kojih podataka, u ovom slučaju vrsta tumora, zastupljeno u skupu podataka. Klase tumora sadrže približno jednak broj slika (oko 826), dok je slika bez tumora samo 395. Programski kod za ispisivanje broja slika u odgovarajućoj mapi za treniranje:

#### Odsječak koda 1 - ispisivanje broja slika u svakoj klasi skupa za treniranje

```
meningeoma_tumor_train = len(os.listdir('/content/drive/MyDrive/Diplomski/
MR brain tumor/Training/meningioma_tumor'))
glioma_tumor_train = len(os.listdir('/content/drive/MyDrive/Diplomski/MR b
rain tumor/Training/glioma_tumor'))
pituitary_tumor_train = len(os.listdir('/content/drive/MyDrive/Diplomski/M
R brain tumor/Training/pituitary_tumor'))
no_tumor_train = len(os.listdir('/content/drive/MyDrive/Diplomski/MR brain
tumor/Training/no_tumor'))

print("Number of images for each class (training)")
print("\nmeningeoma tumor:" , meningeoma_tumor_train)
print("glioma tumor:" , glioma_tumor_train)
print("pituitary tumor:" , pituitary_tumor_train)
print("no tumor:" , no_tumor_train)
print("-----")
print("sum:" , meningeoma_tumor_train + glioma_tumor_train + pituitary_tum
or_train + no_tumor_train)

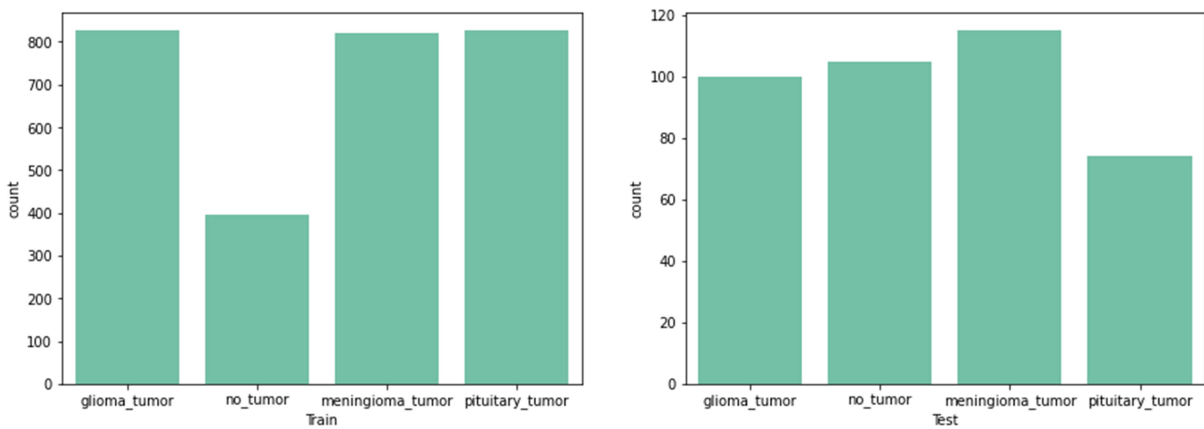
Number of images for each class (training)

meningeoma tumor: 822
glioma tumor: 826
pituitary tumor: 827
no tumor: 395
-----
sum: 2870
```

Skup za testiranje sadrži sveukupno 394 slika, od toga 115 slika meningeoma, 100 slika glioma, 74 tumora hipofize te 105 slika bez tumora.

Skup za treniranje podijeljen je na skup za treniranje i skup za validaciju te su spremljeni u prethodno napravljene direktorije za trening i validaciju, dok je skup za testiranje spremljen u direktorij za testiranje. Unutar istih napravljene su direktoriji za svaku klasu.

Za postizanje što boljih rezultata treniranja modela poželjno je da svaka klasa sadrži približno isti broj slika. Na grafikonu je vidljivo da raspodjela nije približno jednaka jer je stupac klase slika bez tumora znatno niži od ostalih. Ukoliko se trenira model na skupu podataka s neravnomjernom raspodjelom, model će većinu podataka za validaciju i testiranje klasificirati kao klasu s najvećim brojem slika, odnosno klasu koja ima najmanje slika neće dobro naučiti, pa samim time niti prepoznati na testnim i validacijskim skupovima. Takav model nema zadovoljavajuće rezultate te bi korištenje takvog modela u praksi moglo imati loše posljedice.



Slika 30. grafikon raspodjele podataka

Tehnikom augmentacije bilo je potrebno povećati broj slika bez tumora koje su spremljene u istoimenu mapu skupa za treniranje.

Prije izgradnje modela potrebno je izvršiti pripremu podataka kako bi model postigao što bolje rezultate. Alatom ImageDataGenerator definirani su generatori podataka za treniranje, validaciju i testiranje koji učitavaju slike iz direktorija. Pomoću generatora je određena dimenzija slike i veličina serije kako bi treniranje bilo moguće na više slika odjednom (u ovom slučaju na 50 slika).

```
Found 2973 images belonging to 4 classes.
Found 285 images belonging to 4 classes.
Found 394 images belonging to 4 classes.
```

Slika 31: skup za testiranje sadrži 2973, skup za validaciju 285, a skup za testiranje 394 slika raspoređenih u 4 klase

## 7.2. Izrada i treniranje modela

Za mogućnost izrade modela u potpunosti samostalno ispočetka, učitava se sekvencijalni model (engl. sequential model), a kod osobnog modela za klasifikaciju tumora je sljedeći:

Odsječak koda 2 - osobni model za klasifikaciju tumora

```
model = Sequential()
model.add(Conv2D(filters=16, kernel_size=(3,3), padding= 'same',
                 activation='relu', input_shape = (224,224,3)))
model.add(Conv2D(filters=32, kernel_size=(3,3), padding= 'same',
                 activation='relu'))
model.add(MaxPooling2D(pool_size = (2,2), strides = (2,2)))
model.add(BatchNormalization())
model.add(Conv2D(filters=32, kernel_size=(3,3), padding= 'same',
                 activation='relu'))
model.add(MaxPooling2D(pool_size = (2,2), strides = (2,2)))
model.add(BatchNormalization())
model.add(Conv2D(filters=64, kernel_size=(3,3), padding= 'same',
                 activation='relu'))
model.add(MaxPooling2D(pool_size = (2,2), strides = (2,2)))
model.add(BatchNormalization())
model.add(Conv2D(filters=64, kernel_size=(3,3), padding= 'same',
                 activation='relu'))
model.add(MaxPooling2D(pool_size = (2,2), strides = (2,2)))
model.add(BatchNormalization())
model.add(Conv2D(filters=128, kernel_size=(3,3), padding= 'same',
                 activation='relu'))
model.add(MaxPooling2D(pool_size = (2,2), strides = (2,2)))
model.add(BatchNormalization())
model.add(Conv2D(filters=128, kernel_size=(3,3), padding= 'same',
                 activation='relu'))
model.add(MaxPooling2D(pool_size = (2,2), strides = (2,2)))
model.add(Flatten())
model.add(Dense(units = 512, activation='relu'))
model.add(Dense(units = 4, activation='softmax'))
model.summary()
```

S obzirom na mali broj podataka, za bolje rezultate modela, korišteno je nekoliko modela pomoću transferiranog učenja i arhitektura mreža EfficientNet, ResNet50v2, Inceptionv3 te Xception. Stoga nakon definiranja generatora slijedi učitavanje modela koji će biti korišten za klasifikaciju podataka. Korištene su težine s podatkovnog skupa Imagenet, bez potpuno povezanih izlaznih slojeva modela što omogućuje dodavanje i uvježbavanje novog izlaznog sloja, dok su pod `input_shape` specificirane dimenzije slike. U kodu ispod prikazana je funkcija `flatten` koja pretvara sve rezultirajuće 2-dimenzionalne nizove iz skupnih mapa značajki u jedan dugi kontinuirani linearni vektor, funkcija `dropout` kako bi se 50% neurona isključilo te ostali dodani slojevi.

Odsječak koda 3 - slojevi dodani na EfficientNet model

```
model1 = effnet.output
model1 = tf.keras.layers.MaxPooling2D(pool_size = (2,2), strides = (2,2))(
model1)
model1 = tf.keras.layers.Flatten()(model1)
model1 = tf.keras.layers.Dropout(rate=0.5)(model1)
model1 = tf.keras.layers.Dense(units = 4, activation='softmax')(model1)
model1 = tf.keras.models.Model(inputs=effnet.input, outputs = model1)
model1.summary()
```

Često se koristi više povratnih poziva, a u ovom slučaju izabrana su tri. Funkcija `ModelCheckpoint` omogućuje redovito spremanje modela tijekom treniranja na temelju metrike. Model je spremljen prema najboljem iznosu validacijske točnosti. `Tensorboard` služi kako bi se sažetak treniranja modela kasnije mogao vizualizirati te pratiti napredak. `ReduceLRonPLateau` će promijeniti stopu učenja, točnije smanjiti učenje na temelju metrike, kada se metrika prestane poboljšavati.

Odsječak koda 4 - povratni pozivi

```
tensorboard = TensorBoard(log_dir = 'logs')
checkpoint = ModelCheckpoint(filepath_effnet, monitor="val_accuracy",
save_best_only=True,mode="auto",verbose=1)
reduce_lr = ReduceLRonPlateau(monitor = 'val_accuracy', factor = 0.3,
patience = 2, min_delta = 0.001, mode='auto',verbose=1)
```

Da bi se model mogao trenirati potrebno ga je kompilirati: odrediti optimizator i funkciju gubitka te specificirati određene metrike koje će model procijeniti tijekom obuke i testiranja. Odabran je optimizator Adam, za funkciju gubitka je korištena

categorical\_crossentropy funkcija zbog zadatka višeklasne klasifikacije, dok je za evaluaciju od metrika korištena točnost (engl. accuracy).

Odsječak koda 5 - kompiliranje modela

```
model1.compile(loss='categorical_crossentropy',
               optimizer = 'Adam',
               metrics= ['accuracy'])
```

Treniranje modela vrši se putem metode fit kojoj se proslijede argumenti: putanja podataka za treniranje i podataka za validaciju na temelju kojih se procjenjuje gubitak i sve metrike modela na kraju svake epohe, broj epoha (u ovom slučaju dvanaest), verbose definiran na vrijednost jedan, odnosno da prikazuje traku napretka pri učenju i popis povratnih poziva. Veličina serije (engl. batch size) definirana je kod generatora koji generira serije od pedeset koraka. Broj slika koje se treniraju u jednoj epohi jednak je ukupnom broju slika skupa za treniranje podijeljenom sa veličinom serije što u ovom slučaju iznosi šezdeset. Fit funkcija vraća "hystroy" objekt, a njegov atribut je zapis vrijednosti gubitka treniranja i vrijednosti metrike u uzastopnim epohama, kao i vrijednosti gubitka validacije i vrijednosti validacije metrike.

Odsječak koda 6 - funkcija fit za treniranje mreže

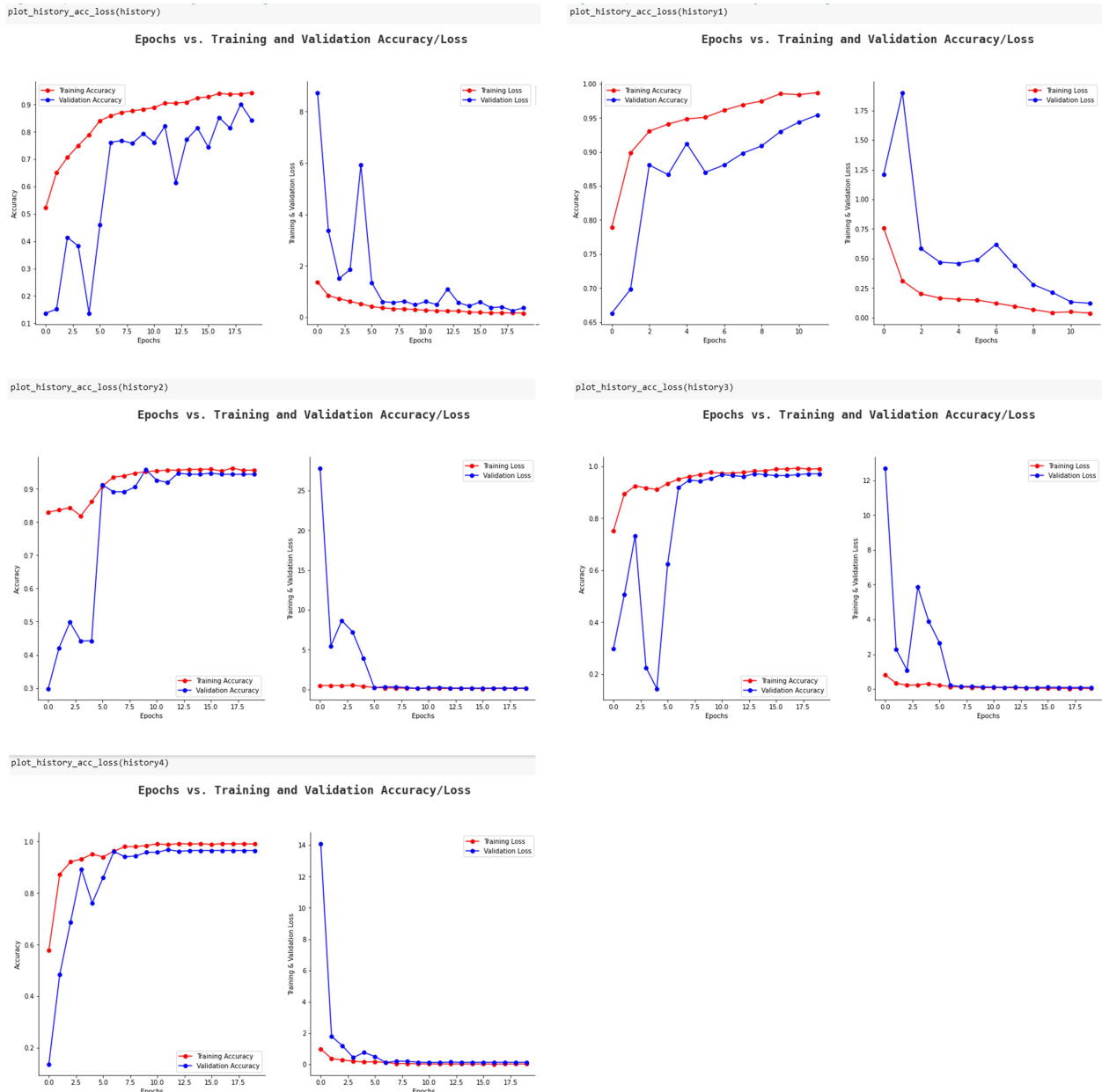
```
history1= model1.fit(train_generator, validation_data=validation_generator,
                    epochs = 12, verbose=1,
                    callbacks=[tensorboard, checkpoint, reduce_lr])
```

Za arhitekture Resnet50v2, Inceptionv3 i Xception postupak pripreme podataka jednak je gore opisanom za EfficientNetB0 arhitekturu i osobni model. Nakon učitavanja, Resnet50v2 je doraden kodom:

Odsječak koda 7 - slojevi dodani na ResNet model

```
model2 = resnet.output
model2 = tf.keras.layers.MaxPooling2D(pool_size = (2,2), strides = (2,2))(
model2)
model2 = tf.keras.layers.Dropout(rate=0.5)(model2)
model2 = tf.keras.layers.BatchNormalization()(model2)
model2 = tf.keras.layers.Dense(units = 256, activation='relu')(model2)
model2 = tf.keras.layers.Flatten()(model2)
model2 = tf.keras.layers.Dense(units = 4, activation='softmax')(model2)
model2 = tf.keras.models.Model(inputs=resnet.input, outputs = model2)
model2.summary()
```

Treniranje mreže možemo vizualizirati pomoću grafova. Vrijednosti na osi x predstavljaju broj epoha, a na osi y nalazi se određena metrika koja je definirana u compile metodi. Na donjoj slici prikazani su grafovi za svaku mrežu.



Slika 32: Grafovi točnosti i gubitka za svaku mrežu redom: osobni model, EfficientNetB0, ResNet50v2, Inception te na posljetku Xception



## 7.3. Evaluacija i testiranje modela

Nakon treniranja modela potrebno je napraviti analizu rezultata. Za osobni model dobiveni loss (rezultat funkcije gubitka, `categorical_crossentropy`) iznosi 0.1574, accuracy (prosjeak točnosti modela) iznosi 94%, validation loss iznosi 0.3611, validation accuracy 84%.

Potom je izvršena evaluacija modela na podacima za testiranje (slika 33). Točnost (engl. accuracy) je mjera koliko je točno predviđanje modela u usporedbi s pravim podacima. Za razliku od točnosti, gubitak (engl. loss) nije postotak. To je zbroj pogrešaka učinjenih za svaki primjer u skupovima za obuku ili validaciju. Vrijednost gubitka pokazuje kako se model ponaša nakon svake iteracije optimizacije i koliko je loše bilo predviđanje modela na jednom primjeru. Cilj uvježbavanja modela je pronaći skup težina i odstupanja koji u prosjeku imaju što niže gubitke u svim primjerima.

Rezultat evaluacije iznosi:

- za osobni model 70,30% točnosti, dok je gubitak 2,609440
- za model EfficientNetB0 80,71% točnosti, dok je gubitak 1,358564
- za ResNet50v2 74,37% točnosti, dok je gubitak 1,437821
- za Inceptionv3 78,68% točnosti, dok je gubitak 1,464637
- za Xception 79,44% točnosti, dok je gubitak 1,437896

```
test_loss1, test_acc1 = model1.evaluate(test_generator)
print("The accuracy of the model is: {:.2f}%".format(test_acc1*100))
print("The loss of the model is: {:.6f}".format(test_loss1))
```

```
13/13 [=====] - 2s 114ms/step - loss: 1.3586 - accuracy: 0.8071
"The accuracy of the model is:": 80.71%
"The loss of the model is:": 1.358564
```

Slika 33: evaluacija modela EfficientNetB0

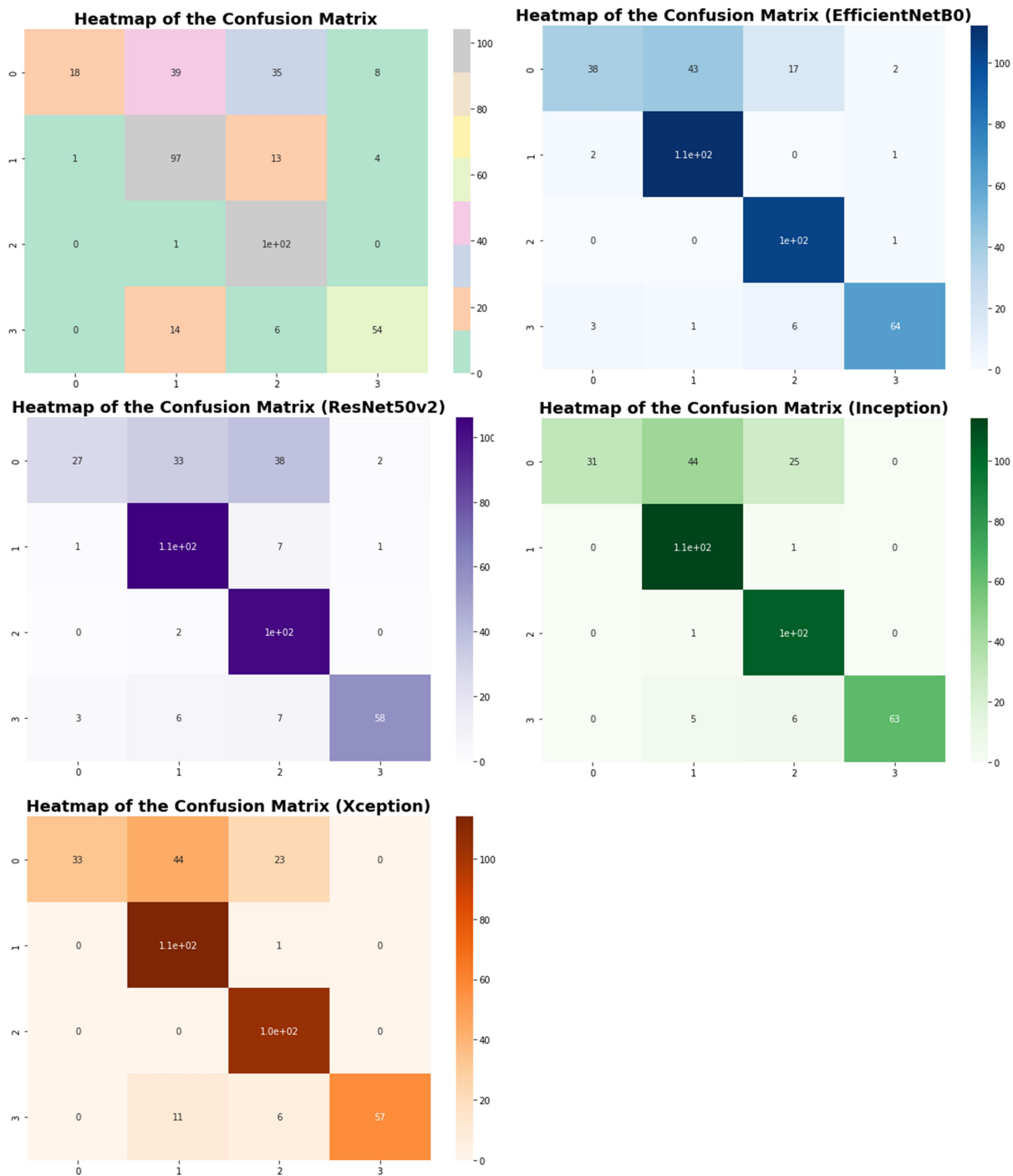
Slijedi predikcija podataka na podacima za testiranje, odnosno testiranje modela na podacima koji nisu označeni u kategorije. Metoda generira izlazna predviđanja za ulazne uzorke. Dakle, za svaku sliku određuje se postotak pripadnosti svakoj klasi, klasi bez tumora, klasi glioma, meningeoma ili klasi tumora hipofize. Sliku klasificiramo u onu kategoriju kojoj model dodijeli najveći postotak. Cilj je postići što veći broj točno klasificiranih slika.

```
print(classification_report(y_test_new, pred1))
```

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| 0            | 0.88      | 0.38   | 0.53     | 100     |
| 1            | 0.72      | 0.97   | 0.83     | 115     |
| 2            | 0.82      | 0.99   | 0.90     | 105     |
| 3            | 0.94      | 0.86   | 0.90     | 74      |
| accuracy     |           |        | 0.81     | 394     |
| macro avg    | 0.84      | 0.80   | 0.79     | 394     |
| weighted avg | 0.83      | 0.81   | 0.78     | 394     |

Slika 34: izvješće o klasifikaciji nakon predikcije modela EfficientNetB0

Nakon završetka predikcije ispisano je izvješće o klasifikaciji (slika 34.) te su rezultati prikazani grafički pomoću heatmap konfuzijske matrice. Konfuzijska matrica prikazuje broj točno i netočno klasificiranih slika za svaku klasu.



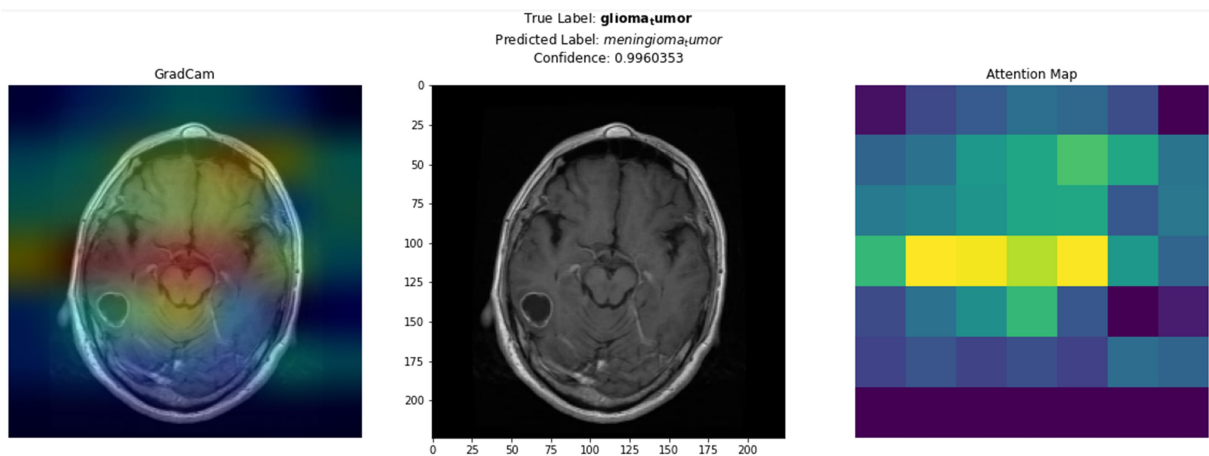
Slika 35: konfuzijska matrica za svaki model

Kako bi se mogla napraviti GradCAM toplinska mapa definirana je funkcija `get_img_array` koja slici dodaje dimenziju za transformaciju niza u seriju veličine (1, 224, 224, 3).

Prva funkcija `gradcam_heatmap` započinje izradom modela koji mapira ulaznu sliku na aktivacije posljednjeg konvolucijskog sloja i izlaza predviđanja te izračunava gradijent najviše predviđene klase za ulaznu sliku s obzirom na aktivacije posljednjeg konvolucijskog sloja. Potom određuje gradijent izlaznog neurona s obzirom na izlaznu mapu značajki posljednjeg konvolucijskog sloja i vektor gdje je svaki unos srednji intenzitet gradijenta preko određenog kanala mape značajki. Svaki kanal u nizu mape značajki se množi prema važnosti kanala s obzirom na najvišu predviđenu klasu, a onda se zbroje svi kanali kako bi dobili aktivaciju klase toplinske mape. U svrhu vizualizacije, također se normalizira toplinska mapa između 0 i 1.

Druga funkcija `display_gradcam` ponovno skalira toplinsku mapu na raspon vrijednosti od 0-255. Koristi `colormap` za bojanje toplinske mape RGB vrijednostima te stvara sliku s RGB toplinskom mapom u boji i superpozicionira toplinsku mapu preko izvorne slike.

Pomoću grafa dobiven je prikaz originalne slike te njezine GradCam toplinske mape s lijeve strane i mape pozornosti s desne strane.



Slika 36: prikaz GradCAM toplinske mape i mape pozornosti MR slike mozga

## 8. Zaključak

U radiološkoj praksi, obučeni liječnici vizualno procjenjuju medicinske slike za otkrivanje, karakterizaciju i praćenje bolesti. Razvoj radiološke opreme, dovodi do bržeg i kvalitetnijeg rada zdravstvenih djelatnika, no ujedno povećava broj učinjenih radioloških pretraga, količinu informacija i nalaza koje radiolozi moraju interpretirati. Zbog ograničenog broja dostupnih obučenih radiologa, povećava se njihovo radno opterećenje. Interpretacijom velikog broja medicinskih slika u malom vremenu, greške su neizbježne. Cilj sustava umjetne inteligencije u radiologiji je pomoći radiolozima u interpretaciji nalaza, odnosno bržoj i kvalitetnijoj dijagnostici. Algoritmi umjetne inteligencije, posebno duboko učenje, izvrsni su u automatskom prepoznavanju složenih obrazaca u slikovnim podacima. Dok su različite arhitekture dubokog učenja istražene za rješavanje različitih zadataka, konvolucijske neuronske mreže (CNN) danas su najraširenije tipologije arhitekture dubokog učenja u medicinskom oslikavanju. Danas modeli daju rezultate visoke točnosti, no problem je što skupovi podataka imaju mali broj slika za učenje mreže te postoje razlike u standardima prikupljanja podataka i protokolima snimanja, osobito među institucijama. Ponekad je modele potrebno ponovno trenirati na novom skupu podataka. Transferirano učenje pomaže u povećanju točnosti modela. U ovom radu, model bez transferiranog učenja postigao je točnost 70%, dok se transferiranim učenjem podigla za desetak posto.

# Literatura

1. Shijun Wang, Ronald M. Summers: Machine Learning and Radiology, 1 Jul 2013  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3372692/>
2. Dinggang Shen, Guorong Wu, Heung-Il Suk: Deep Learning in Medical Image Analysis, 21 Jun 2017 <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5479722/>
3. Deep learning for healthcare: review, opportunities and challenges: R. Miotto, F. Wang, S. Wang, X. Jiang, J.T. Dudley, 06 May 2017  
<https://academic.oup.com/bib/article/19/6/1236/3800524>
4. John R. Haaga, Charles F. Lanzieri: CT and MR imaging of the whole body, 2002.
5. MRI Appearance of Primary Brain Tumors  
[http://www.aboutcancer.com/mri\\_gbm.htm](http://www.aboutcancer.com/mri_gbm.htm)
6. Glioblastoma, IDH-wildtype, Aug 2022  
<https://radiopaedia.org/articles/glioblastoma-idh-wildtype>
7. M. Wang, Z. Wang, P. Ren, X. Zhang, S.L.: Meningioma with ring enhancement on MRI: a rare case report, 10 February 2021  
<https://bmcmimedimaging.biomedcentral.com/articles/10.1186/s12880-021-00555-x>
8. Meningioma: Apr 2022 <https://radiopaedia.org/articles/meningioma>
9. <https://med.virginia.edu/neurosurgery/services/pituitary-tumor-program/pituitary-tumor-frequently-asked-questions/>
10. IBM Cloud Education: Artificial Intelligence (AI), 3 June 2020  
<https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>
11. Eda Kavlakoglu: AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference?, 27 May 2020  
<https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks>
12. IBM Cloud Education: Machine Learning, 15 July 2020  
<https://www.ibm.com/cloud/learn/machine-learning>
13. A. Hosny, C. Parmar, J. Quackenbush, L.H. Schwartz, H.J.W.L. Aerts: Artificial intelligence in radiology, 30 Nov 2018  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6268174/>

14. IBM Cloud Education: Neural Networks, 17 August 2020  
<https://www.ibm.com/cloud/learn/neural-networks>
15. Bojana Dalbelo Bašić, Marko Čupić, Jan Šnajder: Umjetne neuronske mreže, Fakultet elektrotehnike i računarstva, nastavni materijali, svibanj 2008
16. IBM Cloud Education: Recurrent Neural Networks, 14 September 2020  
<https://www.ibm.com/cloud/learn/recurrent-neural-networks>
17. IBM Cloud Education: Convolutional Neural Networks, 20 October 2020  
[https://www.ibm.com/cloud/learn/convolutional-neural-networks?mhsrc=ibmsearch\\_a&mhq=cnn](https://www.ibm.com/cloud/learn/convolutional-neural-networks?mhsrc=ibmsearch_a&mhq=cnn)
18. K. G. van Leeuwen, S. Schalekamp, M. J. C. M. Rutten, B. van Ginneken, M. de Rooij: Artificial intelligence in radiology: 100 commercially available products and their scientific evidence, 15 Apr2021  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8128724/>
19. M. Sirshar, M.F.K. Paracha , M.U. Akram, N.S.Alghamdi, S. Z.Y. Zaidi: Tatheer FatimaAttention based automated radiology report generation using CNN and LSTM, 6 Jan, 2022  
<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0262209>
20. K. Yasaka, O. Abe: Deep learning and artificial intelligence in radiology: Current applications and future directions, 30 Nov 2018  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6267951/>
21. R.Yamashita, M. Nishio, R. K. G. Do, K. Togashi: Convolutional neural networks: an overview and application in radiology, 22 June 2018  
<https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9>
22. IBM Cloud Education: Convolutional Neural Networks, 20 October 2020  
<https://www.ibm.com/cloud/learn/convolutional-neural-networks>
23. CNN Padding, 01.11.2018 <https://datahacker.rs/what-is-padding-cnn/>
24. Ethan Irby: Gradient Descent vs Stochastic GD vs Mini-Batch SGD, 6 Mar 2021  
<https://medium.com/analytics-vidhya/gradient-descent-vs-stochastic-gd-vs-mini-batch-sgd-fbd3a2cb4ba4>
25. Damir Kopljar: Konvolucijske neuronske mreže, završni rad, srpanj 2016.

26. SAGAR SHARMA: Activation Functions in Neural Networks, 6 Sep 2017  
<https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>
27. Artem Oppermann: Underfitting and Overfitting in Deep Learning, 18 Jul 2021  
<https://medium.com/mlearning-ai/underfitting-and-overfitting-in-deep-learning-687b1b7eb738>
28. B. Chen: A Practical Introduction to Early Stopping in Machine Learning, 29 July 2020  
<https://laptrinhx.com/a-practical-introduction-to-early-stopping-in-machine-learning-2456645136/>
29. Normalizing Inputs for an Artificial Neural Network, 20 Jun 2022  
<https://www.baeldung.com/cs/normalizing-inputs-artificial-neural-network>
30. V. Haswani: Learning Rate Decay and methods in Deep Learning, 3 Sep 2020  
<https://medium.com/analytics-vidhya/learning-rate-decay-and-methods-in-deep-learning-2cee564f910b>
31. R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra: Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization, last revised 3 Dec 2019  
<https://arxiv.org/abs/1610.02391>
32. Rachel Draelos: Grad-CAM: Visual Explanations from Deep Networks, 29 Mar 2020  
<https://glassboxmedicine.com/2020/05/29/grad-cam-visual-explanations-from-deep-networks/>
33. Mingxing Tan, Quoc V. Le: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, 11 Sep 2020  
<https://paperswithcode.com/paper/efficientnet-rethinking-model-scaling-for>
34. Aqeel Anwar: Difference between AlexNet, VGGNet, ResNet, and Inception, Jun 7, 2019  
<https://towardsdatascience.com/the-w3h-of-alexnet-vggnet-resnet-and-inception-7baaecccc96>
35. Gaudenz Boesch: Deep Residual Networks (ResNet, ResNet50) – 2022 Guide, 2022.  
<https://viso.ai/deep-learning/resnet-residual-neural-network/>
36. Aakash Kaushik: Understanding ResNet50 architecture  
<https://iq.opengenus.org/resnet50-architecture/>



37. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich: Going deeper with convolutions, 17 Sep 2014  
<https://arxiv.org/abs/1409.4842>
38. Francois Chollet: Xception: Deep Learning with Depthwise Separable Convolutions, July 2017 <https://ieeexplore.ieee.org/document/8099678>
39. Ian Goodfellow, Yoshua Bengio, Aaron Courville. Deep Learning. MIT Press , 2016  
<https://www.deeplearningbook.org/?fbclid=IwAR3U6v INvYvObsAaLw4b1dCMY AwfnjrPI 4CHZGkqHYTqbX8IN84uwmgkA>
40. Keras Documentation <https://keras.io/api/>

## Popis slika

Slika 1: primjeri radioloških snimki različitih modaliteta, izvor:

<https://www.uclahealth.org/radiology/imaging-specialties>

Slika 2: uređaj za magnetsku rezonancu, izvor: [https://www.researchgate.net/figure/MR-system-components-a-Diagram-showing-the-relative-locations-of-the-main-magnet-coils\\_fig1\\_49645994](https://www.researchgate.net/figure/MR-system-components-a-Diagram-showing-the-relative-locations-of-the-main-magnet-coils_fig1_49645994)

Slika 3: usporedba biološkog neurona i neurona umjetne mreže, izvor:

<https://openbooks.lib.msu.edu/neuroscience/chapter/the-neuron/> + Marijana Zekić-

Sušac: Neuronske mreže, Ekonomski fakultet u Osijeku, nastavni materijali

Slika 4: rekurentna mreža (lijevo) i aciklička mreža (desno), izvor:

<https://nerdthecoder.wordpress.com/2019/02/03/recurrent-neural-net/>

Slika 5: rezultati studije: udio regije interesa, modaliteta snimanja i vrste AI zadatka, izvor: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8128724/>

Slika 6: arhitektura konvolucijske neuronske mreže, izvor:

<https://www.upgrad.com/blog/basic-cnn-architecture/>

Slika 7: primjer prelaženja filtera po slici s pomaknom veličine jedan, izvor: Damir Kopljar: Konvolucijske neuronske mreže, završni rad, srpanj 2016.

Slika 8: primjer rada konvolucijskog sloja, izvor: Damir Kopljar: Konvolucijske neuronske mreže, završni rad, srpanj 2016.

Slika 9: primjer nadopunjavanja nulama, izvor: <https://medium.com/@draj0718/zero-padding-in-convolutional-neural-networks-bf1410438e99>

Slika 10: grafovi aktivacijskih funkcija, izvor: <https://www.herongyang.com/Neural-Network/Playground-Impact-of-Activation-Functions.html>

Slika 11: primjer sloja sažimanja maksimalnom vrijednošću, izvor:

<https://www.projectpro.io/article/introduction-to-convolutional-neural-networks-algorithm-architecture/560>

Slika 12: primjer sloja sažimanja prosječnom vrijednošću, izvor:

<https://www.projectpro.io/article/introduction-to-convolutional-neural-networks-algorithm-architecture/560>

Slika 13: grafovi podučenosti, balansa i prenaučenosti, izvor:

<https://docs.aws.amazon.com/machine-learning/latest/dg/model-fit-underfitting-vs-overfitting.html>

Slika 14: Neuronska mreža prije i nakon nestajanja neurona, izvor:

<https://towardsdatascience.com/an-intuitive-explanation-to-dropout-749c7fb5395c>

Slika 15: ilustracija ranog zaustavljanja, izvor: <https://chatbotslife.com/regularization-in-deep-learning-f649a45d6e0>

Slika 16: primjeri augmentacije slike, izvor:

<https://www.semanticscholar.org/paper/Data-Augmentation-for-Brain-Tumor-Segmentation%3A-A-Nalepa-Marcinkiewicz/601106659dc89cb329e8a045ef6ef288fe37b0dd>

Slika 17: prikaz gradijentnog spusta te lokalni i globalni minimum, izvor:

<https://blog.paperspace.com/intro-to-optimization-in-deep-learning-gradient-descent/>

Slika 18: konvergiranje algoritma, izvor: <https://pub.towardsai.net/want-to-optimize-your-model-use-learning-rate-decay-8ab3ce68addc>

Slika 19: princip rada Grad-CAM-a

Slika 20: karakteristike mrežnih arhitektura Xception, ResNet50v2, Inceptionv3 i EfficientNetb0, izvor: <https://keras.io/api/applications/>

Slika 21: Skaliranje modela. (a) primjer je osnovne mreže; (b)-(d) su konvencionalno skaliranje koje povećava samo jednu dimenziju širine, dubine ili razlučivosti mreže; (e) metoda skaliranja koja ravnomjerno skalira sve tri dimenzije s fiksnim omjerom, izvor: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,

<https://paperswithcode.com/paper/efficientnet-rethinking-model-scaling-for>

Slika 22: arhitektura EfficientNetB0 mreže,

izvor: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,

<https://paperswithcode.com/paper/efficientnet-rethinking-model-scaling-for>

Slika 23: koncept preskakanja veza, izvor: <https://iq.opengenus.org/resnet50-architecture/>

Slika 24: arhitekture ResNet modela, izvor: <https://iq.opengenus.org/resnet50-architecture/>

Slika 25: arhitektura Inception mreže, izvor: Going deeper with convolutions,

<https://arxiv.org/abs/1409.4842>

Slika 26: strogo ekvivalentna reformulacija pojednostavljenog Inception modula, izvor:

Xception: Deep Learning with Depthwise Separable Convolutions,

<https://ieeexplore.ieee.org/document/8099678>

Slika 27: "ekstremna" verzija Inception modula, izvor: Xception: Deep Learning with

Depthwise Separable Convolutions, <https://ieeexplore.ieee.org/document/8099678>

Slika 28: Xception arhitektura, izvor: Xception: Deep Learning with Depthwise

Separable Convolutions, <https://ieeexplore.ieee.org/document/8099678>

Slika 29: primjer slike iz svake klase

Slika 30. grafikon raspodjele podataka

Slika 31: skup za testiranje sadrži 2973, skup za validaciju 285, a skup za testiranje 394

slika raspoređenih u 4 klase

Slika 32: Grafovi točnosti i gubitka za svaku mrežu redom: osobni model,

EfficientNetB0, ResNet50v2, Inception te na poslijetku Xception

Slika 33: evaluacija modela EfficientNetB0

Slika 34: izvješće o klasifikaciji nakon predikcije modela EfficientNetB0

Slika 35: konfuzijska matrica za svaki model

Slika 36: prikaz GradCAM toplinske mape i mape pozornosti MR slike mozga

## Sažetak

Radiološko snimanje ima veliki značaj u njezi bolesnika za rano otkrivanje, dijagnosticiranje i liječenje bolesti. Kod tumora i drugih bolesti mozga, magnetska rezonanca (MR) je jedna od najčešćih metoda snimanja. Razlikovanje tumora mozga vrlo je izazovan zadatak u velikoj prostornoj i strukturnoj varijabilnosti okolnog područja tumora mozga. Stoga je svrha ovog rada razviti model koji se temelji na umjetnoj inteligenciji, odnosno konvolucijskoj neuronskoj mreži, da prepozna tri najčešća tipa tumora mozga, gliom, meningeomom i tumor hipofize pomoću slika magnetske rezonance te ispitati četiri arhitekture neuronskih mreža, EfficientNet, ResNet50v2, Inception te Xception. U prvom dijelu rada opisan je princip rada magnetske rezonance te prikaz najčešćih tumora mozga, dok su u drugom obrađeni pojmovi umjetne inteligencije, strojnog i dubokog učenja, umjetne neuronske mreže te njihove primjene u radiologiji. Arhitektura i način rada konvolucijske neuronske mreže objašnjeni su u trećem dijelu. Nakon toga slijedi opis metode razvoja, treniranja i testiranja mreže na skupu podataka sa stranice Kaggle.com te dobiveni rezultati.

Ključne riječi: strojno učenje, radiologija, duboko učenje, umjetna inteligencija, magnetska rezonanca, tumor mozga

# Abstract

Radiological imaging is significant in patient care regarding early disease diagnostics and treatment. With brain tumors and other brain diseases magnetic resonance imaging (MR) is the most frequently used method of imaging. Detecting and distinction of tumor structure within spatial and structural variability of brain tissue is challenging. Purpose of this thesis is to develop a model based on artificial intelligence, more precisely convolutional neural networks to distinguish three most common types of brain tumors: glioma, meningioma and pituitary tumor using magnetic resonance imaging and test four architectural types of neuron network: EfficientNet, ResNet50v2, Inception and Xception. The first part of thesis describes how MR device works and classifies brain tumors. The second part defines concept of AI, machine and deep learning, artificial neuron network and their usage in radiology. Architecture and function of convolutional neural networks are described in the third part. In the final part there are methods of development, training and testing of network made on the dataset from the Kaggle.com and final results.

Keywords: machine learning, radiology, deep learning, artificial intelligence, magnetic resonance imaging, brain tumor