

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1292

**ANALIZA KARDIOLOŠKIH SLIKA
KORIŠTENJEM DUBOKE NEURONSKE MREŽE**

Adam Vuković

Zagreb, lipanj, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1292

Pristupnik: **Adam Vuković (0035235027)**

Studij: Elektrotehnika i informacijska tehnologija i Računarstvo

Modul: Računarstvo

Mentor: akademik prof. dr. sc. Sven Lončarić

Zadatak: **Analiza kardioloških slika korištenjem duboke neuronske mreže**

Opis zadatka:

Kvantitativna analiza slika u medicini je važna jer povećava točnost dijagnostičkih procedura i samim time poboljšava ishod liječenja za pacijente. Procjena stupnja odbacivanja srčanog tkiva kod pacijenata s transplantiranim srcem važno je za provedbu točne i efikasne terapije. Korištenjem dubokog učenja, određivanje je moguće automatizirati ili olakšati kliničku dijagnostiku. U sklopu završnog rada potrebno je proučiti postojeće metode za analizu medicinskih slika temeljenih na dubokom učenju te odabrati jednu od proučenih metoda te ju implementirati. Točnost implementirane metode potrebno je ispitati korištenjem skupa ispitnih slika i usporediti rezultate s drugim metodama.

Rok za predaju rada: 14. lipnja 2024.

Zahvaljujem se svima koji su na bilo koji način pridonijeli nastanku ovog završnog rada.

Na prvom mjestu, želim izraziti duboku zahvalnost svom mentoru, akademiku prof. dr. sc. Svenu Lončariću te dr. sc. Doniku Vršnaku za savjete, smjernice i razumijevanje kroz cijeli proces istraživanja i pisanja rada.

Posebna zahvala Iani Adamović za pomoć u razumijevanju medicinske terminologije i problematike iznesene u ovom radu.

Hvala mojoj obitelji na podršci i razumijevanju tijekom mog obrazovanja.

Sadržaj

1. Uvod	1
2. Umjetne neuronske mreže	2
2.1. Biološki neuron	2
2.2. Umjetni neuron	3
2.3. Vrste umjetnih neurona	4
2.4. Princip rada umjetne neuronske mreže	6
2.5. Konvolucijske neuronske mreže	11
3. X-PCI slike biopsija transplantiranih srca	14
3.1. Stupanj odbacivanja presatka "0R"	14
3.2. Stupanj odbacivanja presatka "1R"	15
3.3. Stupanj odbacivanja presatka "2R"	15
3.4. Stupanj odbacivanja presatka "3R"	16
3.5. Artefakti	16
4. Programska implementacija analize X-PCI slika	18
4.1. Učitavanje podatkovnog skupa	18
4.2. Modeli neuronskih mreža	20
4.3. Treniranje neuronske mreže	22
5. Rezultati i rasprava	26
5.1. Točnosti i gubitci	26
5.2. Matrice zabune	29
5.3. Područja aktivacije	32
5.4. Interpretacija rezultata i ograničenja	35

6. Zaključak	36
Literatura	38
Sažetak	40
Abstract	41

1. Uvod

Transplantacija srca predstavlja jedan od najkompleksnijih medicinskih postupaka, pružajući novu šansu za život pacijentima s terminalnom srčanom insuficijencijom. Glavni izazov nakon transplantacije je imunološko odbacivanje presatka, što zahtijeva redovito praćenje i preciznu dijagnozu. Tradicionalne metode analize biopsija transplantiranih srca uključuju vizualni pregled uzoraka tkiva pod mikroskopom, što može biti subjektivno i zahtijeva visoko specijaliziranu ekspertizu.

Napredak u umjetnoj inteligenciji, posebno u razvoju dubokih neuronskih mreža, otvara nove mogućnosti za automatizaciju i poboljšanje analize medicinskih slika. Modeli poput DenseNet i ResNet pokazali su izvanredne rezultate u klasifikaciji slika, te bi se njihovom primjenom mogla omogućiti preciznija i brža dijagnoza medicinskih slika. U kontekstu transplantacije srca, primjena ovih modela može značajno unaprijediti proces detekcije i klasifikacije stupnjeva odbacivanja presatka, identificiranih kao 0R, 1R, 2R i 3R.

Cilj ovog rada je istražiti učinkovitost DenseNet i ResNet modela u analizi slika biopsija transplantiranih srca te usporediti njihove performanse u zadacima klasifikacije i regresije. Rad će prikazati opću metodologiju primjene ovih modela i evaluirati njihove performanse u određivanju stupnja odbacivanja presatka.

Ovaj rad će također razmotriti izazove i ograničenja primjene dubokih neuronskih mreža u analizi biopsija te pružiti uvid u potencijalne smjerove za buduća istraživanja.

2. Umjetne neuronske mreže

Razvoj tehnologije u području umjetne inteligencije i strojnog učenja omogućio je kreiranje sustava koji mogu oponašati složene obrasce ljudskog razmišljanja. Jedna od najznačajnijih inovacija unutar ovog polja su umjetne neuronske mreže, koje su inspirirane radom ljudskog mozga. Ove mreže predstavljaju pokušaj da se simuliraju procesi obrade informacija u biološkim neuronima s ciljem stvaranja inteligentnih sustava sposobnih za učenje i prilagodbu.

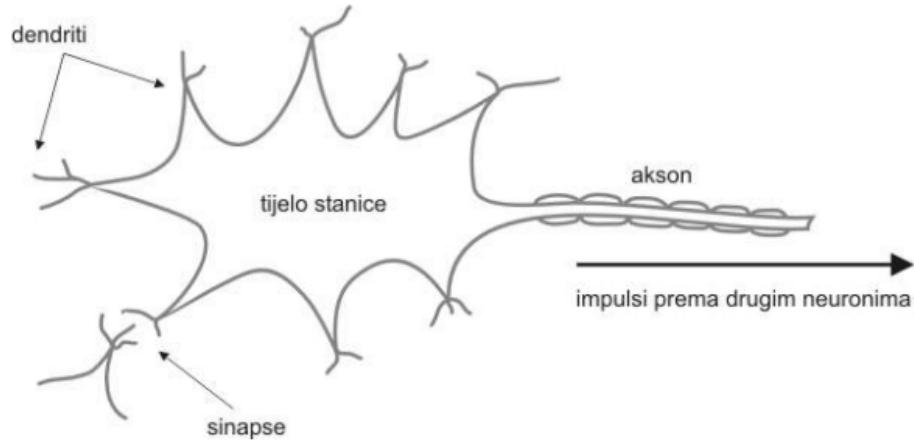
Glavna motivacija iza razvoja umjetnih neuronskih mreža leži u njihovoj sposobnosti da obrađuju velike količine podataka, prepoznaju obrasce te donose odluke na temelju tih podataka, slično načinu na koji biološki mozgovi to rade. Iako je biološki mozak nevjerojatno kompleksan i neponovljiv u svojoj potpunosti, umjetne neuronske mreže se temelje na osnovnim principima rada neurona i sinaptičkih veza, omogućujući stvaranje funkcionalnih modela za primjenu u različitim područjima poput prepoznavanja govora, slike, predviđanja tržišnih trendova te mnogih drugih.

Kroz ovo poglavlje istražit ćemo osnovne elemente umjetnih neuronskih mreža, njihovu strukturu, načine učenja te njihove primjene u stvarnim problemima. Također ćemo raspraviti izazove s kojima se suočavaju, uključujući problem pretreniranosti i iščežavajućeg gradijenta, koji su ključni u optimizaciji performansi ovih modela.

2.1. Biološki neuron

Biološki neuron je osnovna jedinica ljudskog mozga, sastavljen od tijela stanice (soma), dendrita (koji primaju signale od drugih neurona), aksona (koji prenosi električne signale) i završnih članaka (koji prenose signale drugim neuronima). Ljudski mozak sadrži oko 100 milijardi neurona, svaki povezan s tisućama drugih neurona, stvarajući složenu

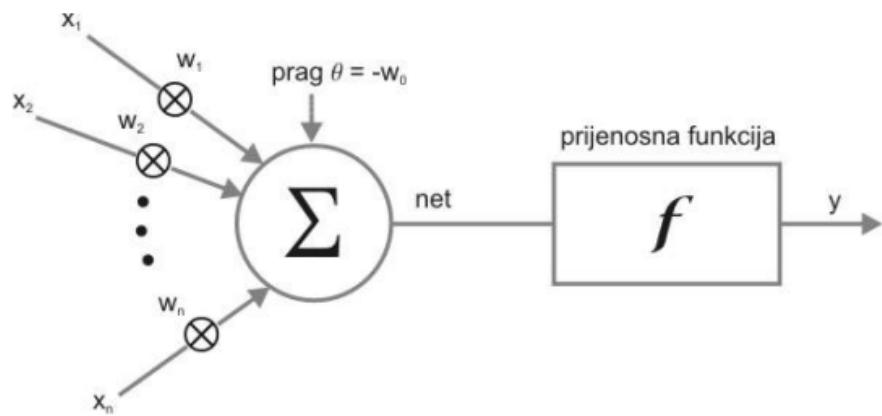
mrežu za obradu informacija.



Slika 2.1. Biološki neuron [1]

2.2. Umjetni neuron

Umjetni neuron, prema McCulloch-Pitts modelu, predstavlja pojednostavljenu verziju biološkog neurona. On prima ulazne signale, množi ih s pripadajućim težinama, zbraja rezultate i prolazi kroz aktivacijsku funkciju koja određuje izlaz neurona. Ovaj model omogućuje simulaciju osnovnih procesa učenja i obrade informacija koji se odvijaju u biološkim neuronima.



Slika 2.2. Umjetni neuron [1]

2.3. Vrste umjetnih neurona

Vrste umjetnih neurona razlikuju se po aktivacijskim funkcijama koje koriste. Ovisno o aktivacijskoj funkciji, izlaz neurona, pa tako i ponašanje cijele neuronske mreže, može biti sasvim drugačije.

Prvo razmotrimo najjednostavniju aktivacijsku funkciju: funkciju identiteta. Ta funkcija je jednaka samoj težinskoj sumi ulaza, dakle matematički je definirana kao

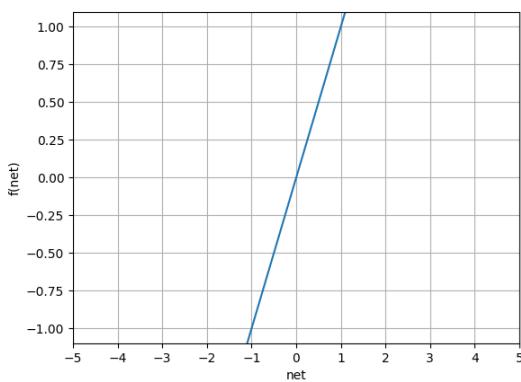
$$f(\text{net}) = \text{net} \quad (2.1)$$

Tip neurona koji koristi ovakvu aktivacijsku funkciju zove se ADALINE neuron (*Adaptive Linear Element*).

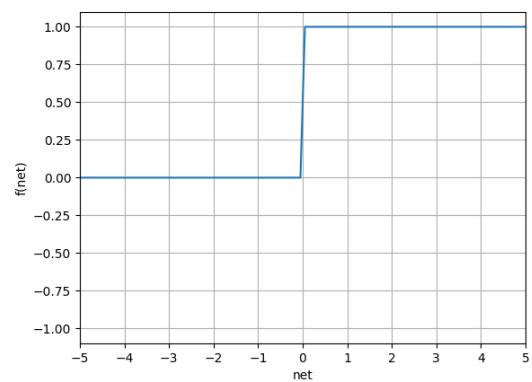
Iduća funkcija koju ćemo razmotriti je funkcija skoka koja imitira ponašanje aktivacije biološkog neurona koji se aktivira samo ako pređe određeni prag (u funkciji je određeno da je prag jednak nuli):

$$f(\text{net}) = \text{step}(\text{net}) = \begin{cases} 0 & \text{za } \text{net} < 0 \\ 1 & \text{inače} \end{cases} \quad (2.2)$$

Tip neurona koji koristi ovakvu aktivacijsku funkciju zove se TLU (*Threshold Linear Unit*).



1. Funkcija identiteta



2. Funkcija skoka

Iduće dvije funkcije su preinake prošle funkcije. Razlikuju se od funkcije skoka po

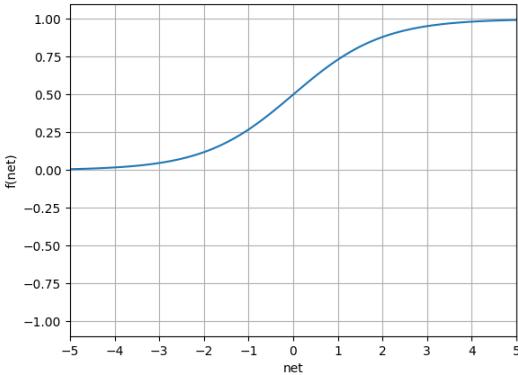
tome što nemaju točan prag kada se neuron aktivira, već vrijednost izlaza postaje kontinuirana. Izlazi asimptotski teže prema donjoj, odnosno gornjoj granici.

Prva od te dvije aktivacijske funkcije zove se sigmoidalna funkcija i definirana je kao

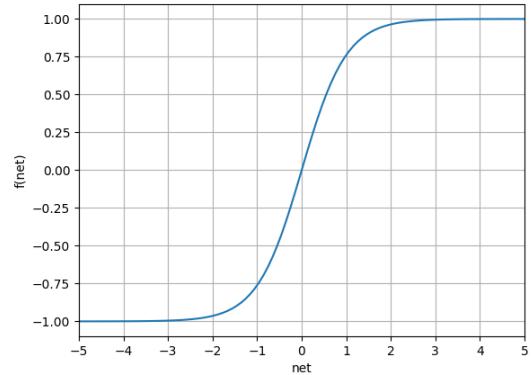
$$f(\text{net}) = \text{sigm}(\text{net}) = \frac{1}{1 + e^{-\text{net}}} \quad (2.3)$$

Za drugu aktivacijsku funkciju ovog tipa koristi se funkcija tangens hiperbolni

$$f(\text{net}) = \tanh(\text{net}) = \frac{2}{1 + e^{-2 \cdot \text{net}}} = 2 \cdot \text{sigm}(2 \cdot \text{net}) - 1 \quad (2.4)$$



3. Sigmoidalna funkcija



4. Funkcija tangens hiperbolni

Radi neučinkovitosti prethodno spomenutih aktivacijskih funkcija kod rješavanja određenih problema, noviji modeli koriste neurone zglobnice (*ReLU*, *Rectified Linear Unit*) i propusne zglobnice (*Leaky ReLU*). Ovaj tip funkcija koristi ideje svih triju prethodnih tipova funkcija i kombinira ih u jednu.

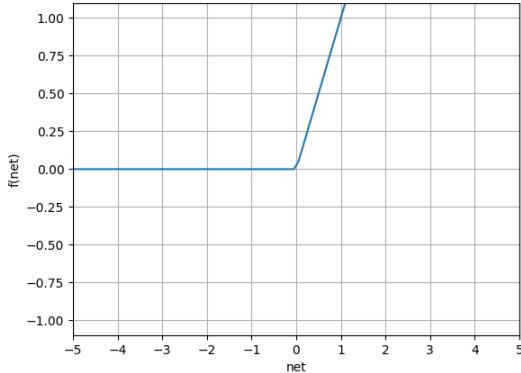
Funkcija aktivacije zglobnice glasi

$$f(\text{net}) = \max(0, \text{net}) \quad (2.5)$$

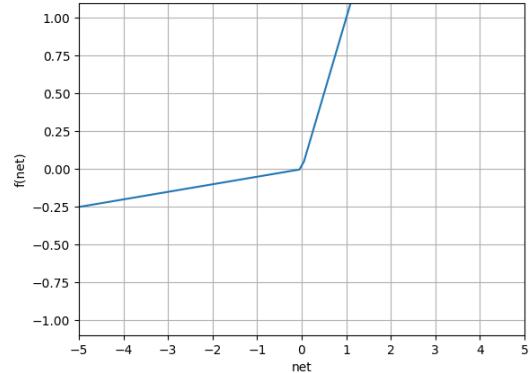
Funkcija aktivacije propusne zglobnice još dodaje parametar α koji dopušta izlazu neurona da padne ispod nule za ulaze koji su manji od nule, dok je vrijednost za ulaze

koji su veći ili jednaki nuli isti kao i kod zglobnice

$$f(\text{net}) = \begin{cases} \text{net} & \text{za } \text{net} \geq 0 \\ \alpha \cdot \text{net} & \text{inače} \end{cases} \quad (2.6)$$



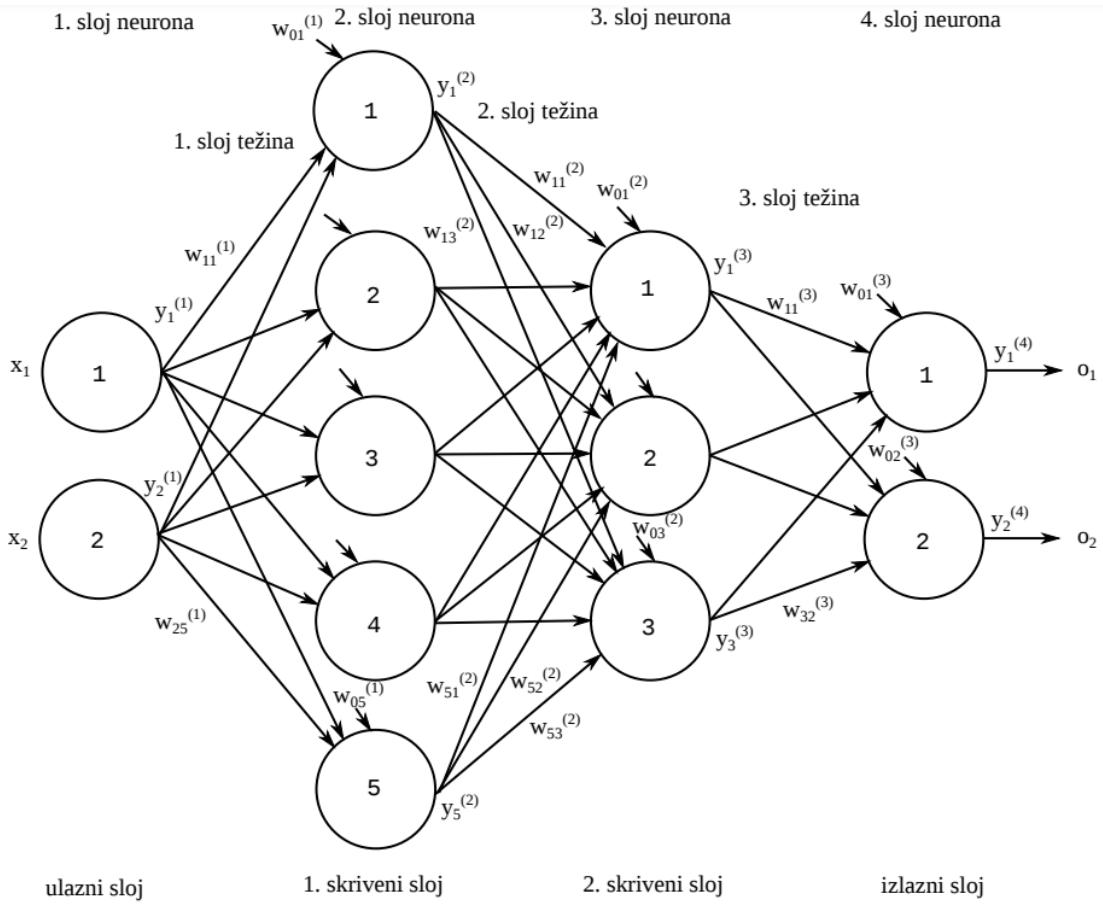
5. Zglobnica



6. Propusna zglobnica uz $\alpha = 0.05$

2.4. Princip rada umjetne neuronske mreže

Neuronske mreže sastoje se od slojeva međusobno povezanih neurona. Svaki neuron prima ulazne podatke, množi ih s pripadajućim težinama, zbraja rezultate i prolazi kroz aktivacijsku funkciju kako bi proizveo izlaz. U dubokim neuronskim mrežama, koje se sastoje od više skrivenih slojeva, ovaj proces omogućuje modelu da uči hijerarhije značajki iz podataka. Najjednostavnija arhitektura umjetnih neuronskih mreža je unaprijedna neuronska mreža (*Feedforward Neural Network*) u kojoj su slojevi povezani na način da informacija teče samo u jednom smjeru – od ulaza prema izlazu.



Slika 2.6. Arhitektura unaprijedne neuronske mreže [2]

Neuronske mreže se koriste za rješavanje raznih zadataka, od kojih su najčešći klasifikacija i regresija. Klasifikacija se odnosi na predviđanje diskretnih kategorija, poput prepoznavanja objekata na slikama ili, u našem slučaju, klasifikacije stupnja odbacivanja presatka. Regresija, s druge strane, uključuje predviđanje kontinuiranih vrijednosti, poput predikcije vremenskih uvjeta.

Rad s neuronским mrežama odvija se kroz dvije faze: fazu učenja i fazu obrade podataka. Faza učenja je iterativan postupak gdje mreži predajemo podatke iz skupa za učenje (*training set*) na kojima mreža optimizira svoje parametre. Učenje mreže može biti nadzirano (*supervised*), nenadzirano (*unsupervised*) i podržano (*reinforcement*). Nadzirano učenje optimizira svoje parametre na temelju ulaznih podataka i ciljanih izlaza, nenadzirano učenje ne dobiva ciljane izlaze tako da je ono dobro za traženje pravilnosti u podacima, a podržano učenje traži optimalne strategije na temelju pokušaja s odgođenom nagradom. Faza učenja također može analizirati skup za provjeru (*validation set*)

koji pokazuje uspješnost mreže na neviđenim podacima u svrhu testiranja generalizacije modela. Svaka iteracija ove faze zove se epoha. U fazi obrade podataka mreža se koristi treniranim modelom kako bi predvidjela izlaze na temelju novih, neviđenih podataka iz skupa za testiranje (*test set*).

Nakon prikazane šire slike učenja neuronskih mreža, potrebno je objasniti nekoliko izraza i algoritama koji će pobliže objasniti na koji način neuronska mreža zapravo uči, tj. optimizira svoje parametre. Radi jednostavnosti u nastavku je opisan način na koji uči unaprijedna neuronska mreža prilikom nadziranog učenja.

Kako bi neuronska mreža učila na što učinkovitiji način, bitno je kvantificirati koliko je predviđeno rješenje mreže daleko od stvarnog rješenja. To radimo uz pomoć funkcije gubitka. Postoji mnogo funkcija gubitaka, a u nastavku su definirane dvije koje su korištene u programskoj implementaciji ovog rada.

Prva funkcija gubitka koja je korištena zove se srednje kvadratno odstupanje (*MSE, mean squared error*) koja je definirana kao

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.7)$$

gdje y_i predstavlja stvarne oznake, a \hat{y}_i predviđene oznake podataka. Broj uzoraka označen je s n . Ova funkcija gubitka najčešće se koristi kod regresijskih modela.

Druga funkcija gubitka koja je korištena zove se gubitak unakrsne entropije (*cross entropy loss*) koja je definirana kao

$$L = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^k y_{ij} \log(\hat{y}_{ij}), \quad (2.8)$$

gdje y_{ij} predstavlja stvarne oznake podataka (1 ako primjer i pripada klasi j , inače 0), a \hat{y}_{ij} predstavlja predviđene vjerojatnosti da primjer i pripada klasi j . Broj klasa označen je s k , a broj uzoraka je opet označen kao n . Ova funkcija gubitka koristi se isključivo u klasifikacijskim modelima.

Algoritam propagacije unazad (*backpropagation*) je tehnika koja se koristi za izračunavanje gradijenta funkcije gubitka u odnosu na težine u mreži. Nakon što mreža

proizvede izlaz, pogreška se računa na temelju funkcije gubitka. Ta pogreška se zatim propagira unatrag kroz mrežu sve do ulaznog sloja. Parcijalnim deriviranjem svakog parametra možemo izračunati gradijent pogreške.

Izračunati gradijent je ključan u optimiziranju parametara mreže pomoću gradijentnog spusta. Cilj gradijentnog spusta je minimizirati funkciju gubitka. Gradijentni spust ažurira težine neurona tako što računa smjer najbržeg pada funkcije gubitka i pomiče se u tom smjeru. Matematika iza gradijentnog spusta može se opisati jednadžbom:

$$\theta := \theta - \alpha \cdot \nabla J(\theta) \quad (2.9)$$

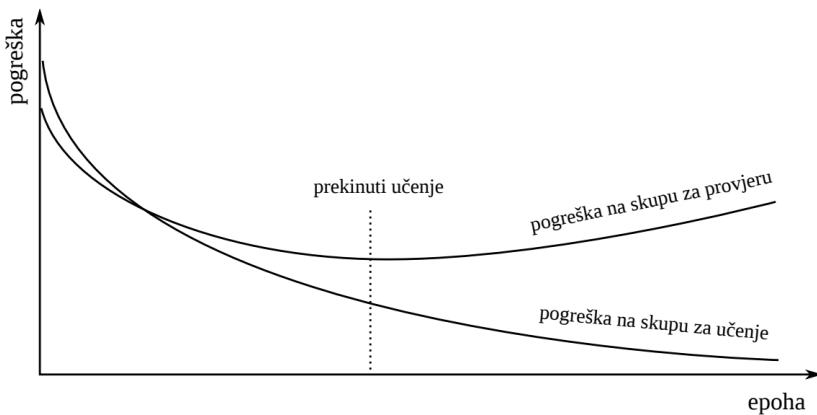
gdje je θ vektor težina, α stopa učenja, a $\nabla J(\theta)$ gradijent funkcije gubitka. Stopa učenja je proizvoljno odabrani parametar koji označava veličinu koraka prilikom gradijentog spusta. Taj parametar mora biti manji od 1 te bi trebao biti obrnuto proporcionalan veličini skupa za trening. Ako je stopa učenja prevelika, mreža može imati problema kod traženja minimuma. Suprotno, ako je stopa učenja premala, faza učenja može potrajati predugo.

Postoje tri tipa gradijentnog spusta temeljena na odnosu brzine epohe i učinkovitosti ažuriranja parametara. Prvi tip je stohastički gradijentni spust (*stochastic gradient descent*) gdje mreža ažurira svoje parametre nakon svakog predočenog uzorka. Taj tip uzrokuje najbržu epohu, ali učinkovitost ažuriranja parametara je loša zbog mogućnosti šumovitih gradijenata. S druge strane, gradijentni spust u grupama (*batch gradient descent*) uzima sumu svih pogrešaka u skupu za učenje i zatim ažurira parametre mreže. Ovaj tip jako učinkovito ažurira parametre, ali trajanje epohe traje jako dugo. Treći i najkorišteniji tip je gradijentni spust u minigrupama (*mini-batch gradient descent*) koji balansira brzinu epohe i učinkovitost ažuriranja parametara. On radi na način da se skup za učenje podijeli u više mini-grupa pa da se ažuriranje provodi za svaku mini-grupu. Veličina mini-grupe ovisi o resursima s kojim programer raspolaže.

Iako se čini da je poznavanjem gore navedenih koncepata lako napraviti efikasnu neuronsku mrežu, to u stvarnosti nije tako. Potrebno je upoznati se i s problemima kod učenja neuronskih mreža. U nastavku je objašnjeno nekoliko najčešćih problema.

Pretreniranost (*overfitting*) se događa kada mreža previše precizno uči obrasce iz skupa

za učenje, uključujući i šum ili specifične karakteristike koje se ne pojavljuju u stvarnom svijetu. Kao rezultat toga, mreža loše generalizira na nove podatke. S druge strane, podtreniranost (*underfitting*) nastaje kada model nije dovoljno kompleksan da nauči osnovne obrasce iz podataka, što dovodi do loših performansi na svim skupovima podataka. Na slici je prikazana pogreška kroz epohe s označenim trenutkom kad bismo trebali prekinuti učenje i spremiti optimalni model s kojim ćemo analizirati slike iz skupa za testiranje.



Slika 2.7. Pogreška kroz epohe [2]

Iščežavajući gradijent (*vanishing gradient*) se događa kada gradijenti prilikom algoritma propagacije unazad u početnim slojevima mreže postanu izrazito mali pa računalo te vrijednosti spremi kao 0. To se najčešće javlja kod dubokih neuronskih mreža koje koriste aktivacijske funkcije čije su derivacije manje od 1, primjerice sigmoidalna i funkcija tangens hiperbolni. Rezultat ovog problema je sporije ili nikakvo učenje u početnim slojevima. Rješenje problema moglo bi biti korištenje aktivacijske funkcije čija derivacija nije manja od 1, primjerice zglobnica.

U suprotnosti, eksplodirajući gradijent (*exploding gradient*) se događa kada gradijenti prilikom algoritma propagacije unazad u početnim slojevima mreže postanu izrazito veliki pa računalo te vrijednosti spremi kao NaN. To se događa kada su derivacije aktivacijskih funkcija veće od 1, ili kada su težine u mreži inicijalizirane s prevelikim vrijednostima. Rezultat ovog problema je destabilizacija procesa učenja i oscilaciranje ili divergiranje modela, što onemogućuje njegovu konvergenciju. Rješenje problema uključuje smanjenje dimenzionalnosti mreže, obrezivanje gradijenta i bolja inicijalizacija težina.

2.5. Konvolucijske neuronske mreže

Konvolucijske neuronske mreže (*CNN, convolutional neural networks*) su posebna vrsta dubokih neuronskih mreža koje su posebno učinkovite za analizu slika. Njihova arhitektura i način rada omogućuju automatsko učenje prostornih hijerarhija značajki iz ulaznih slika. CNN-ovi se sastoje od nekoliko ključnih tipova slojeva koji zajedno omogućuju prepoznavanje obrazaca i klasifikaciju slika. U nastavku ćemo detaljno opisati osnovne komponente CNN-ova: ulazni sloj, konvolucijski sloj, sloj sažimanja i potpuno povezani sloj. Na kraju potpoglavlja opisane su dvije suvremene arhitekture CNN-ova koje se koriste u programskoj implementaciji ovoga rada.

Ulazni sloj (*input layer*) CNN-a prihvata sirove podatke slike, koje obično predstavljamo kao trodimenzionalne tenzore. Prva dimenzija tenzora predstavlja visinu slike, druga širinu slike, a treća predstavlja broj kanala. Ako se radi o slici u boji (*RGB*), tada je broj kanala jednak 3 za svaku od crvene, zelene i plave boje. Ako se radi o crno-bijeloj slici (*grayscale*), tada je broj kanala jednak 1.

Prije objašnjavanja konvolucijskog sloja, potrebno je objasniti samu operaciju konvolucije. Konvolucija je proces koji kombinira dvije funkcije da bi se proizvela treća. Matematički, konvolucija između slike, koju predstavljamo kao $f(t)$ i filtera, kojeg predstavljamo kao $g(t)$ definira se kao

$$y(t) = (f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau, \quad (2.10)$$

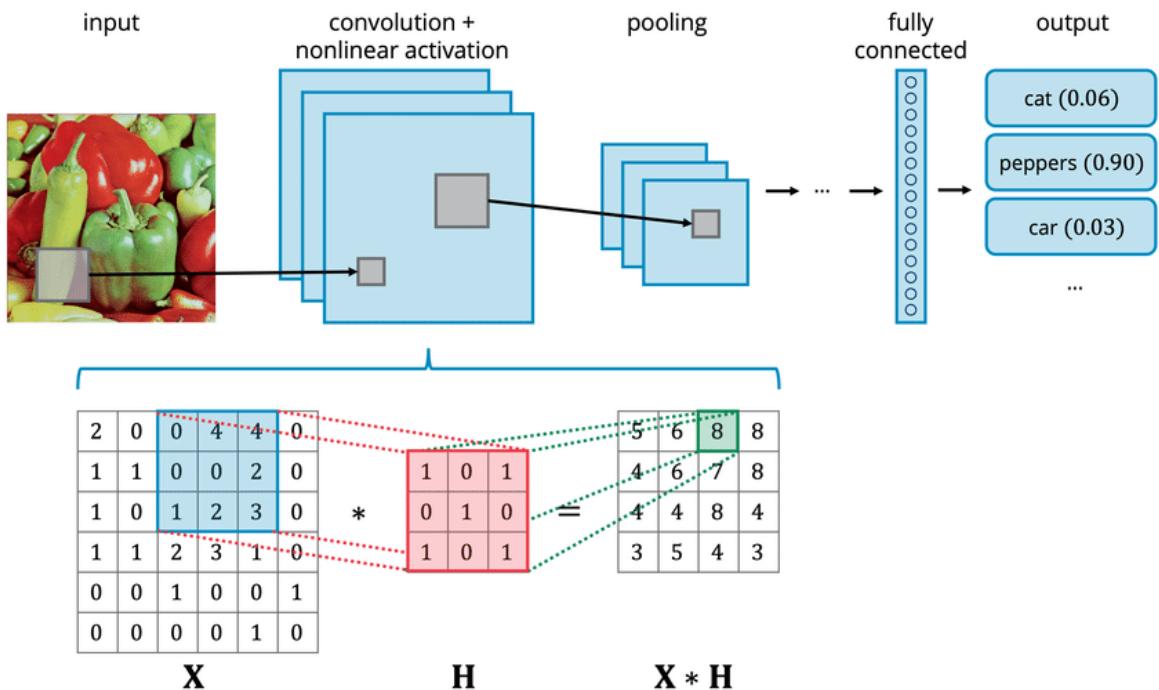
gdje su $f(t)$ i $g(t)$ neprekidne funkcije.

Konvolucijski sloj (*CONV, convolutional layer*) je ključna komponenta CNN-a koja primjenjuje konvolucijske filtre na ulazne podatke. Svaki konvolucijski filter (*kernel*) je mala matrica težina (često dimenzija 3×3) koji se pomiče preko ulazne slike i izračunava skalarne proizvode između težina filtera i preklapajućih regija ulazne slike. Rezultat je mapa značajki (*feature map*) koja naglašava različite značajke slike, kao što su rubovi, teksture i oblici.

Sloj sažimanja (*POOL, pooling layer*) smanjuje dimenzionalnost mapa značajki dok zadržava važne informacije. Najčešće korišteni tip sažimanja je *max-pooling*, koji uzima

maksimalnu vrijednost iz svake podregije mape značajki. Drugi tip je *average-pooling*, koji računa prosječnu vrijednost u podregiji.

Potpuno povezani sloj (*FC, fully connected layer*) je završni sloj CNN-a koji povezuje sve neurone iz prethodnog sloja sa svakim neuronom u ovom sloju. Ovaj sloj obavlja klasifikacijske zadatke na visokoj razini na temelju značajki ekstrahiranih u prethodnim slojevima. Aktivacija neurona računa se težinskom sumom izlaza prethodnog sloja baš kao i kod klasičnih neuronskih mreža.

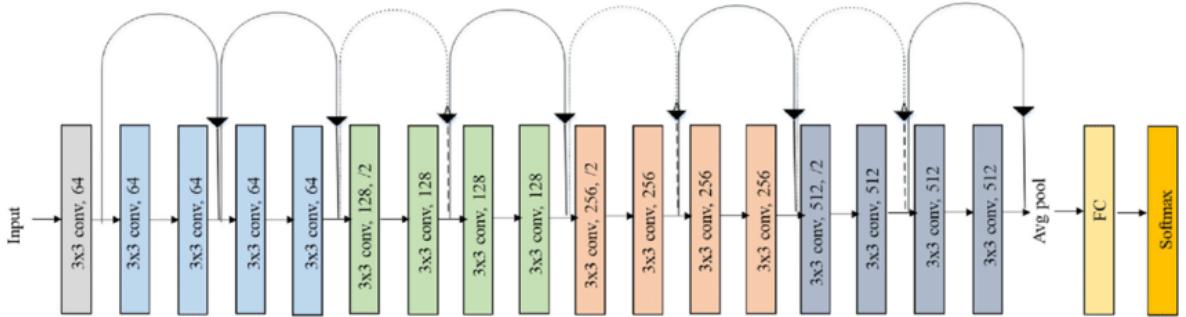


Slika 2.8. Arhitektura i primjer rada konvolucijske neuronske mreže [3]

U programskoj implementaciji ovoga rada korištene su dvije suvremene i popularne arhitekture: ResNet i DenseNet. U nastavku ćemo pobliže opisati obje arhitekture.

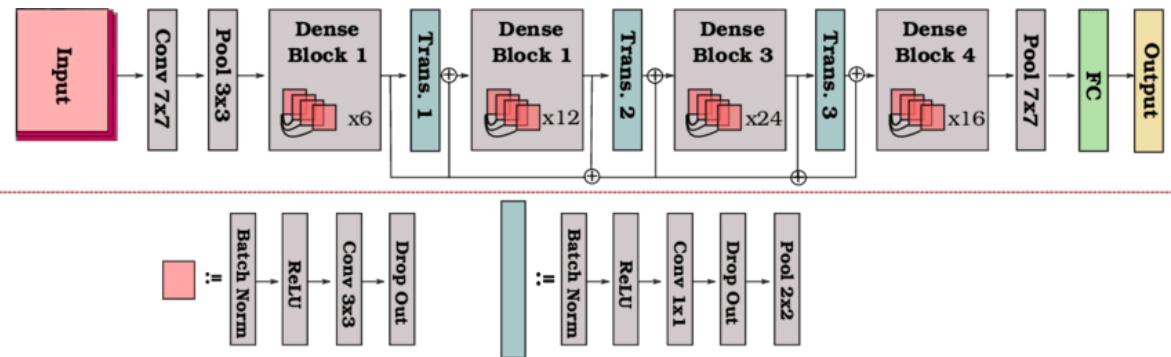
ResNet ili rezidualna mreža je arhitektura razvijena 2015. godine od strane Microsofta. Ona je donijela poboljšanje u odnosu na prijašnje duboke neuronske mreže koje su imale problema s eksplodirajućim ili isčežavajućim gradijentom. Problem je riješen implementacijom rezidualnih blokova koji koriste tehniku zvanu *skip connections* koja povezuje jedan sloj s drugim dalnjim slojem preskačući nekoliko slojeva između. Na taj način je stvoren rezidualni blok (*residual block*). Takvi blokovi su učinkoviti jer mogu preskakati slojeve koji štete učenju neuronske mreže. ResNet-18 arhitektura je vrsta rezidualnih mreža koja ima 18 slojeva. Za implementaciju naše analize korištena je upravo

ova vrsta zbog usporedbe s puno dubljom DenseNet-121 mrežom koja je obrađena u nastavku.



Slika 2.9. ResNet-18 arhitektura [4]

DenseNet je arhitektura predstavljena 2016. godine u radu *Densely Connected Convolutional Networks* Gao Huanga, Zhuang Liua, Laurens van der Maatena i Kilian Q. Weinbergera. Ona koristi guste veze (*dense connections*) koje spajaju svaki sloj s prethodnim slojevima unutar gustog bloka (*dense block*). To znači da svaki sloj prima kao ulaz konkatenirane mape značajki svih prethodnih slojeva unutar tog bloka. Implementacija gustih blokova uvelike pomaže s problemom iščežavajućeg gradijenta. DenseNet također koristi prijenosni sloj (*transition layer*) kako bi kontrolirao veličinu mape značajki te sloj uskog grla (*bottleneck layer*) koji smanjuje broj parametara. Sloj uskog grla omogućava da DenseNet arhitekture, unatoč većem broju slojeva, u pravilu imaju manje parametara od ResNet arhitektura. Označavajući broj slojeva mreže kao L , umjesto da mreža ima L veza kao tradicionalna CNN ili $2L$ veza kao uobičajena ResNet mreža, ova arhitektura nam omogućava da imamo $L(L + 1)/2$ veza. U programskoj implementaciji iskorištena je DenseNet-121 arhitektura koja 4 gusta bloka koji se sveukupno sastoje od 121 sloja.



Slika 2.10. DenseNet-121 arhitektura [5]

3. X-PCI slike biopsija transplantiranih srca

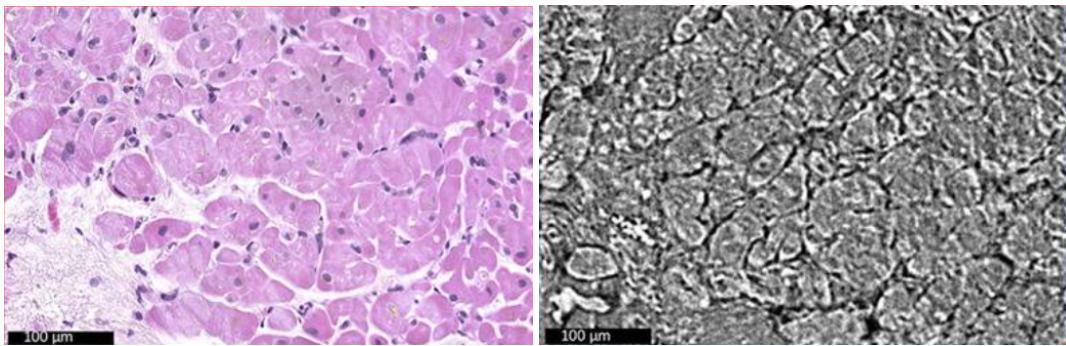
Zlatni standard detekcije akutnog staničnog odbacivanja transplantata je histološka analiza uzorka dobivenih endomiokardijalnom biopsijom. *X-ray phase contrast imaging (X-PCI)* je novija metoda koja omogućava 3D histopatološki prikaz uzorka dobivenog endomiokardijalnom biopsijom bez klasičnog procesa pripreme histološkog preparata.

Histološki, akutno stanično odbacivanje manifestira se infiltracijom miokardnog tkiva bijelim krvnim stanicama, pretežno mononuklearnim stanicama (T- ili B-limfociti, i monociti/makrofagi), i posljedicama infiltracije poput oštećenja kardiomiocita i mikrovaskulature.

Prema dokumentu Međunarodnog društva za transplantaciju srca i pluća (*ISHLT*) iz 2005. godine, predložen je pojednostavljeni sustav ocjenjivanja koji akutno stanično odbacivanje kategorizira u 4 stupnja: 0R, 1R, 2R i 3R.

3.1. Stupanj odbacivanja presatka "0R"

U stupnju 0R nema akutnog staničnog odbacivanja. Stanična infiltracija među stanicama miokarda nije vidljiva, jedine jezgre koje su vidljive su neimunološkog podrijetla (npr. fibroblasti, endotelne stanice). Stupanj 0R nije potrebno liječiti. Na slici 3.1. prikazane su karakteristike ovog stupnja.



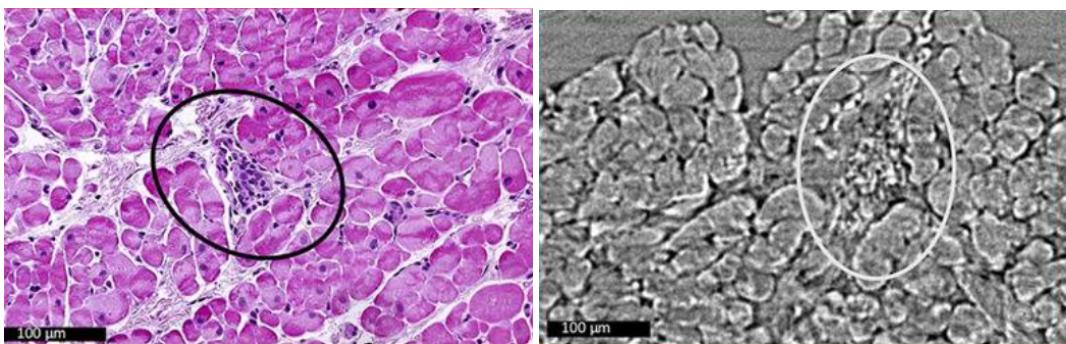
(a) Konvencionalna histologija

(b) X-PCI slika

Slika 3.1. Uvećane histološke slike stupnja "0R" [6]

3.2. Stupanj odbacivanja presatka "1R"

Blago akutno stanično odbacivanje je stupanj 1R. Ono je karakterizirano intersticijskom ili perivaskularnom mononuklearnom infiltracijom. Stupanj 1R nije potrebno liječiti. Na slici 3.2. prikazane su karakteristike ovog stupnja.



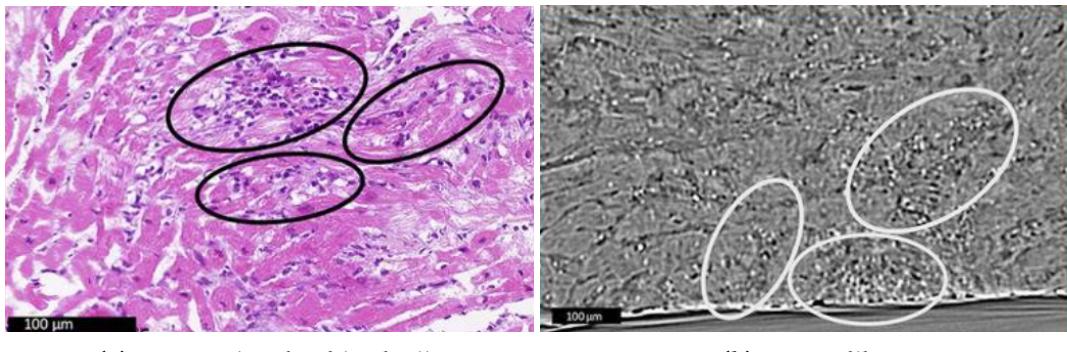
(a) Konvencionalna histologija

(b) X-PCI slika

Slika 3.2. Uvećane histološke slike stupnja "1R" [6]

3.3. Stupanj odbacivanja presatka "2R"

Stupanj 2R je umjereno akutno stanično odbacivanje. Ono je obilježeno istim karakteristikama kao u stupnju "1R", ali na više područja u uzorku. Stupanj 2R je potrebno liječiti imunosupresivnom terapijom. Na slici 3.3. prikazane su karakteristike ovog stupnja.



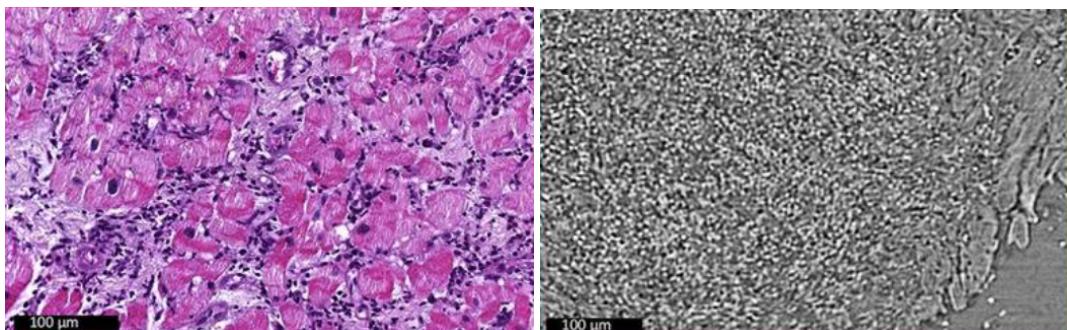
(a) Konvencionalna histologija

(b) X-PCI slika

Slika 3.3. Uvećane histološke slike stupnja "2R" [6]

3.4. Stupanj odbacivanja presatka "3R"

Teško akutno stanično odbacivanje je stupanj 3R. Karakterizira ga difuzna infiltracija mononuklarnih stanica na više područja u uzorku s posljedičnim oštećenjem stanica. Stupanj 3R potrebno je što prije početi liječiti adekvatnom imunosupresivnom terapijom kako bi se proces odbacivanja zaustavio. Na slici 3.4. prikazane su karakteristike ovog stupnja.



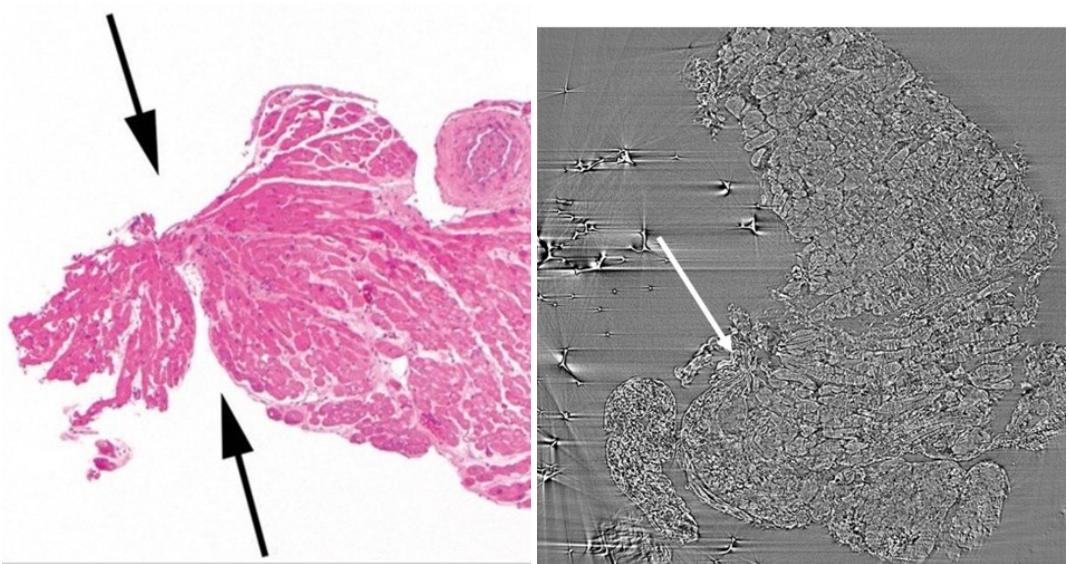
(a) Konvencionalna histologija

(b) X-PCI slika

Slika 3.4. Uvećane histološke slike stupnja "3R" [6]

3.5. Artefakti

U analizi implementiranih modela neuronskih mreža bitno je znati da je na nekim slikama moguće vidjeti artefakte koji nastaju zbog nepravilnog rukovanja uzorkom te isti ne smiju biti prepoznati kao posljedica odbacivanja ili nekog drugog patološkog procesa. Artefakti mogu otežati ili u potpunosti onemogućiti patološku dijagnozu.



(a) Konvencionalna histologija [7]

(b) X-PCI slika

Slika 3.5. Slike artefakta

4. Programska implementacija analize X-PCI slika

U ovom poglavlju opisat ćemo proces programske implementacije analize X-PCI slika, koja se temelji na korištenju naprednih tehnika dubokog učenja. Implementacija se u velikoj mjeri oslanja na radni okvir PyTorch, koji je jedan od najpopularnijih i najmoćnijih alata za razvoj i treniranje neuronskih mreža.

PyTorch, razvijen od strane Meta AI-a, korisnicima omogućava jednostavno definiranje i optimizaciju modela neuronskih mreža. Njegova fleksibilnost i dinamika omogućuju brzo iteriranje i eksperimentiranje s različitim arhitekturama modela, što je od iznimne važnosti u radu s kompleksnim slikovnim podacima poput X-PCI slika.

Poglavlje je strukturirano u nekoliko potpoglavlja koja će se fokusirati na različite aspekte implementacije. U potpoglavlju 4.1. bit će objašnjeno kako se vrši učitavanje i priprema podatkovnog skupa. Potpoglavlje 4.2. pokriva implementaciju arhitektura DenseNet i ResNet neuronskih mreža, dok se potpoglavlje 4.3. bavi procesom treniranja tih modela, uključujući metode optimizacije i evaluacije performansi.

4.1. Učitavanje podatkovnog skupa

U ovom potpoglavlju opisujemo postupak učitavanja i pripreme podatkovnog skupa korištenjem prilagođene klase `CustomImageDataset` koja omogućava pretraživanje direktorija kako bi se pronašle slike biopsija transplantiranih srca, filtriranje slika prema zadanim kriterijima, te primjenu odgovarajućih transformacija slika za treniranje, provjeru i testiranje neuronske mreže.

Podatkovni skup sastoji se od crno-bijelih X-PCI slika iz Kliničko bolničkog centra Zagreb u .tif formatu koje su organizirane u direktorije prema oznakama stupnja odba-

civanja presatka: 0R, 1R, 2R, i 3R. Svaki direktorij ima otprilike dvije tisuće slika koje predstavljaju visokokvalitetan videozapis presjeka srca. Gledajući ime direktorija, slijedila je vrlo lako dati oznaku stupnja koja nam treba za nadzirano učenje mreže. Upravo to je još jedan od zadataka prilagođene klase `CustomImageDataset`.

Transformacija slika ostvarena je uz pomoć modula `transforms` iz biblioteke `torchvision`. Kako bismo spriječili prenaučenost neuronske mreže, u skupu za treniranje dodane su transformacije nasumičnog rotiranja slika. Nakon toga, dodano je i pretvaranje slike u tenzor kako bi neuronska mreža mogla dobiti sliku u brojčanom obliku. Dimenzije tenzora su jednake rezoluciji slike (1024×1024), vrijednosti su između 0 (crna boja) i 1 (bijela boja). Naposlijetu, primijenjena je transformacija normalizacije s očekivanom vrijednošću 0.5 i standardnom devijacijom 0.5 kako bi neuronska mreža lakše našla tražene značajke. Skupovi za provjeru i testiranje koriste samo zadnje dvije transformacije, s time da je skupu za testiranje na početak dodana još jedna koja postavlja rezoluciju slike na 1024×1024 jer su slike iz tog skupa izvorne rezolucije 2560×2560 .

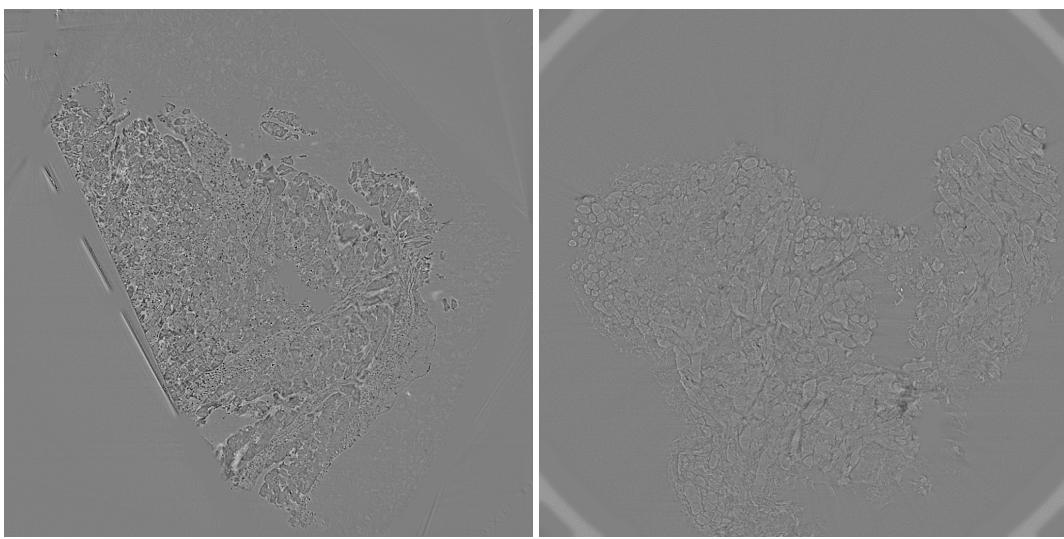
Nakon instanciranja sva tri objekta klase `CustomImageDataset`, konačno možemo napraviti tri objekta klase `DataLoader` iz modula `torch.utils.data`. Neuronskoj mreži ćemo davati podatke u minigrupama (*mini-batches*). Minigrupe su potrebne kako bi razvijeni model mogao bolje generalizirati. Premda je uobičajeno koristiti veće minigrupe, u našoj implementaciji minigrupa će sadržavati 4 slike zbog ograničenih resursa.

Podatkovni skup za treniranje sastoji se od biopsija 9 srca, dok skup za provjeru ima biopsije 4 srca. Skup za testiranje ima biopsije 14 srca. Distribucija stupnjeva po skupovima dana je u nastavku:

Stupanj	Skup za treniranje	Skup za provjeru	Skup za testiranje
0R	8012	2160	4221
1R	2160	2160	4721
2R	4320	2160	4919
3R	3881	2160	8138

Tablica 4.1. Distribucija broja slika po skupovima

Bitno je napomenuti da su tkiva iz skupova za treniranje i provjeru preparirana u formalinu, a to se na slikama očituje kao jači kontrast između segmenata što pomaže neuronskoj mreži da lakše uoči tražene segmente. S druge strane, taj je proces dugačak i neučinkovit tako da su slike podatkovnog skupa za testiranje dobivene direktno iz biopsija, bez preparacije u formalinu. U nastavku prikazana je slika tkiva prepariranog u formalinu te slika tkiva koje nije preparirano u formalinu za stupanj "3R" jer se u tome stadiju najbolje vide segmenti koje neuronska mreža treba pronaći.



(a) Skup za treniranje/provjeru

(b) Skup za testiranje

Slika 4.1. Usporedba kontrasta u podatkovnim skupovima (stupanj "3R")

4.2. Modeli neuronskih mreža

Prijenosno učenje (*transfer learning*) je tehnika gdje se model umjetne neuronske mreže treniran za jedan zadatak koristi kao početna točka za treniranje na drugom zadatku. Ova metoda iskorištava znanje stečeno na prvom zadatku kako bi poboljšala performanse i smanjila vrijeme treniranja na drugom zadatku. Naša implementacija analize X-PCI slika koristit će upravo tu tehniku implementiranjem četiriju modela: DenseNet klasifikacijski model, DenseNet regresijski model, ResNet klasifikacijski model te ResNet regresijski model.

U našoj implementaciji napravljene su prilagođene klase `ResNetNeuralNetwork` i `DenseNetNeuralNetwork` koje primaju hiperparametar `num_classes` koji određuje broj klasa. Ako želimo regresijski model, broj klasa moramo instancirati na 1. U konstruk-

toru se moramo pobrinuti da prvi konvolucijski sloj prima ulaz s jednim kanalom umjesto ulaza s tri kanala jer koristimo crno-bijele slike umjesto RGB slika. Još jedna izmjena je dodavanje novog sloja s linearom regresijom ili klasifikacijom, ovisno o num_classes, umjesto potpuno povezanog sloja. Metoda za proslijedivanje (*forward method*) u slučaju regresije koristi sigmoidalnu aktivacijsku funkciju čiji je izlaz skaliran na raspon $(-0.5, 3.5)$ kako bi odgovarao stupnju odbacivanja presatka. Ako je model klasifikacijski, tada koristimo softmax funkciju za skaliranje izlaza na raspon $[0, 1]$ za svaku klasu tako da je suma svih izlaza klase jednaka 1.

Listing 4..1: Prilagođena ResNet klasa

```

import torch
import torch.nn as nn
import torchvision.models as models

class ResNetNeuralNetwork(nn.Module):
    def __init__(self, num_classes):
        super(ResNetNeuralNetwork, self).__init__()
        self.base_model = models.resnet18(weights='DEFAULT')
        self.base_model.conv1 = nn.Conv2d(1, 64, kernel_size=(7, 7),
                                       stride=(2, 2), padding=(3, 3), bias=True)
        num_ftrs = self.base_model.fc.in_features
        self.base_model.fc = nn.Identity()
        self.num_classes = num_classes
        self.head = nn.Linear(num_ftrs, num_classes)

    def forward(self, x):
        x = self.base_model(x)
        output = self.head(x)
        if self.num_classes == 1:
            output = torch.sigmoid(output)
            output = 4 * output - 0.5
        else:
            output = torch.softmax(output, dim=1)
        return output

```

Listing 4..2: Prilagođena DenseNet klasa

```

import torch
import torch.nn as nn
import torchvision.models as models

class DenseNetNeuralNetwork(nn.Module):
    def __init__(self, num_classes=1):
        super(DenseNetNeuralNetwork, self).__init__()
        self.base_model = models.densenet121(weights='DEFAULT')
        self.base_model.features[0] = nn.Conv2d(1, 64, kernel_size=(7,
            7), stride=(2, 2), padding=(3, 3), bias=True)
        self.base_model = nn.Sequential(*list(self.base_model.children
            ())[:-1])
        self.num_classes = num_classes
        self.head = nn.Linear(1024, num_classes)

    def forward(self, x):
        x = self.base_model(x)
        x = x.mean([2, 3])
        output = self.head(x)
        if self.num_classes == 1:
            output = torch.sigmoid(output)
            output = 4 * output - 0.5
        else:
            output = torch.softmax(output, dim=1)
        return output

```

4.3. Treniranje neuronske mreže

Nakon implementacija modela neuronskih mreža, preostaje nam samo istrenirati ih. Na početku biraju se parametri poput broja klasa koji određuje i funkciju gubitka. Ako je broj klasa jednak 1, to znači da koristimo regresiju i tada je funkcija gubitka srednje kvadratno odstupanje koja je definirana formulom 2.7 Ako je broj klasa jednak 4, tada koristimo klasifikaciju, a funkcija gubitka je gubitak unakrsne entropije (*cross entropy loss*) koja je definirana formulom 2.8

Još je bitno odabrati stopu učenja koju proslijedujemo optimizatoru kao parametar. Prilikom eksperimenetiranja s različitim stopama učenja, ispostavilo se da je optimalna

vrijednost 10^{-5} za Adam optimizator. Adam optimizator jedan je od najpopularnijih optimizatora zbog svoje učinkovitosti i brzine konvergencije. On kombinira prednosti dvaju drugih optimizatora: AdaGrad (koji radi dobro s rijetkim podacima) i RMSProp (koji radi dobro u nestacionarnim okruženjima).

Prije treniranja modela, bitno je prebaciti model na grafički procesor (*GPU, graphics processing unit*) kako bi treniranje bilo brže te da nam ne istroši resurse. U našoj implementaciji koristimo GPU s CUDA (*Compute Unified Device Architecture*) podrškom. CUDA je platforma za paralelno računanje i programersko sučelje koje je razvila NVidia. Nakon ovoga, petlja treniranja i provjere može početi. Pokazalo se da je dovoljno 15 epoha za optimalan model. U nastavku je prikazan pseudokod petlje za treniranje:

Algorithm 1 Treniranje modela

```
1: lista_gubitaka, lista_točnosti ← Ø
2: for epoha ← 0 to broj_epoha do
3:     tekući_gubitak, točne_predikcije ← 0
4:     for slike, oznake in skup_za_treniranje do
5:         premjesti_na_GPU(slike, oznake)
6:         očisti_gradijente()
7:         predikcije ← unaprijedni_prolaz(slike)
8:         gubitak ← funkcija_gubitka(predikcije, oznake)
9:         gubitak.prosljedi_grešku_unazad()
10:        ažuriraj_optimizator()
11:        tekući_gubitak ← tekući_gubitak + gubitak
12:        if klasifikacija then
13:            predikcije ← argmax(predikcije)
14:        else if regresija then
15:            predikcije ← zaokruži_predikcije(predikcije)
16:        end if
17:        nove_točne_predikcije ← izračunaj_točne_predikcije(predikcije, oznake)
18:        točne_predikcije ← točne_predikcije + nove_točne_predikcije
19:    end for
20:    gubitak ← tekući_gubitak / veličina_skupa_za_treniranje
21:    točnost ← točne_predikcije / veličina_skupa_za_treniranje * 100      ▷ postotak
22:    lista_gubitaka.dodaj(gubitak)
23:    lista_točnosti.dodaj(točnost)
24: end for
```

Listu gubitaka i točnosti spremamo kako bismo nakon izvršavanja svih epoha mogli na grafu prikazati kako se ponašaju njihove krivulje kroz epohe.

Petlja za provjeru vrlo je slična prethodnoj, ali ipak postoji nekoliko razlika. Naočitija stvar u izvedbi je ta da ne moramo pratiti gradijente pošto mreža ne uči na skupu za provjeru. To ostvarujemo uz pomoć funkcionalnosti iz PyTorch paketa: `with torch.no_grad():`. Time znatno ubravamo ovu petlju (u našem slučaju je petlja za provjeru bila

otprilike 3 puta brža od petlje za trening). U petlji za provjeru također pamtimo liste svih predviđenih i stvarnih oznaka kako bi ih kasnije mogli prikazati u matricama zabilješke. Zadnja stvar koju trebamo implementirati je pamćenje najbolje točnosti modela na skupu za provjeru te, ako je točnost najveća dosad, zapamtiti model uz pomoć napisane funkcije `save_checkpoint`. Ona pamti parametre koji nam služe za spremanje performansi modela poput gubitka, točnosti, stvarnih i predviđenih oznaka te parametre koji nam služe za samo spremanje neuronske mreže kako bismo ju mogli koristiti u budućnosti. Ti parametri su stanje modela i optimizator.

5. Rezultati i rasprava

U ovom poglavlju prikazani su rezultati dobiveni analizom modela i njihovih performansi. Cilj je bio evaluirati učinkovitost predloženih modela dubokog učenja u zadatku analize kardioloških *X-PCI slika*.

Rezultati su prezentirani kroz nekoliko ključnih dijelova: usporedba točnosti i gubitaka različitih arhitektura neuronskih mreža, evaluacija regresijskih i klasifikacijskih modela te analiza performansi na skupu za provjeru, kao i na testnom skupu podataka. Također su prikazani grafički prikazi koji omogućuju bolji uvid u ponašanje modela tijekom treniranja i testiranja.

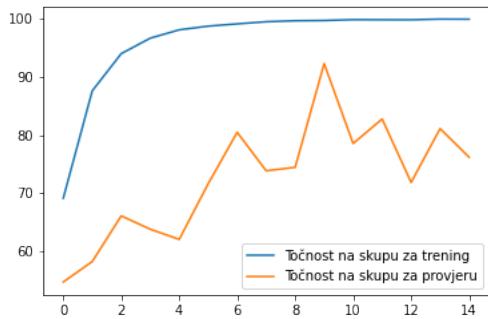
Također, posvećena je i pažnja interpretaciji dobivenih rezultata i njihovom značaju u kontekstu primjene modela u stvarnim kliničkim scenarijima. Na kraju poglavlja, sumirani su ključni nalazi koji pružaju temelje za daljnju raspravu o ograničenjima korištenja dubokih neuronskih mreža u medicinskoj dijagnostici.

5.1. Točnosti i gubitci

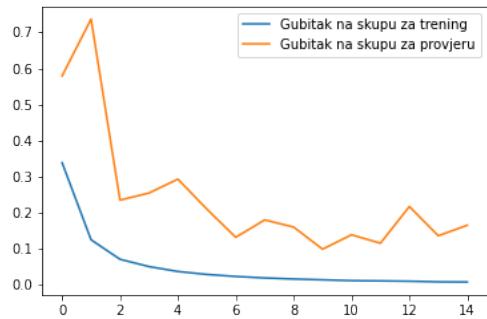
Najosnovnije vrijednosti kojima provjeravamo performanse modela umjetnih neuronskih mreža su točnost i gubitak. Točnost definira postotak ispravno klasificiranih primjera u odnosu na ukupan broj primjera u skupu podataka, a gubitak je definiran funkcijom gubitka u potpoglavlju 2.4.

Slika 5.1. prikazuje krivulje točnosti i gubitka za sve modele, što je u skladu s očekivanjima - točnost modela se povećava, dok se gubitak smanjuje. Rezultati na skupu za provjeru su očekivano lošiji, a krivulje su nestabilnije s izraženim oscilacijama, što je uobičajeno kada model procjenjuje slike koje nije prethodno vido. Vidljivo je da su točnosti i gubitci već od prve epohe na visokoj razini, što je rezultat primjene tehnike

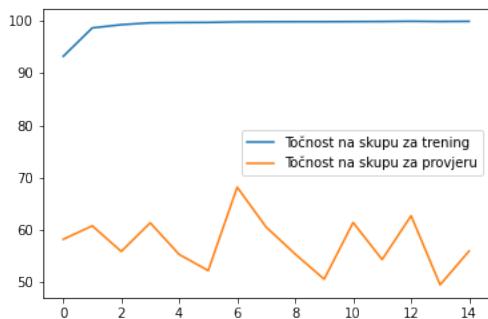
prijenosnog učenja. Ova tehnika koristi prethodno naučene značajke iz već treniranih modela, čime se ubrzava konvergencija i poboljšavaju početni rezultati. Budući da su ResNet i DenseNet trenirani na klasifikacijskim zadacima, točnosti i gubitci su bolji kod klasifikacijskih modela na skupu za trening. Međutim, regresijski modeli pokazuju bolju generalizaciju, što se očituje u njihovim boljim rezultatima na skupu za provjeru.



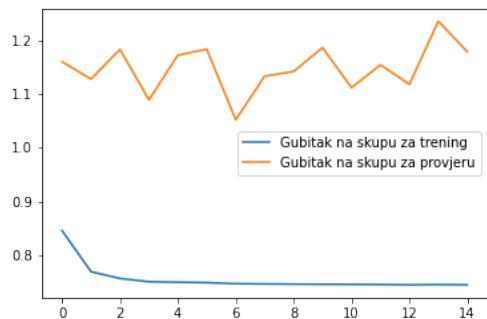
1. ResNet regresija - točnost kroz epohe



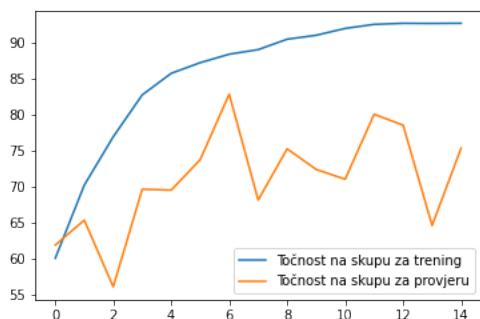
2. ResNet regresija - gubitak kroz epohe



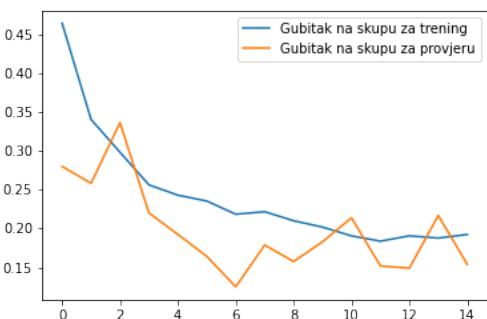
3. ResNet klasifikacija - točnost kroz epohe



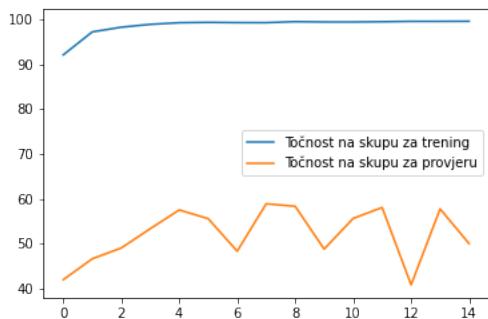
4. ResNet klasifikacija - gubitak kroz epohe



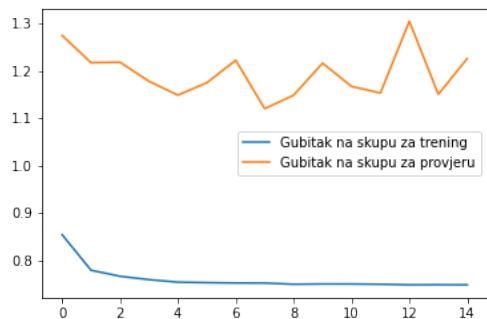
5. DenseNet regresija - točnost kroz epohe



6. DenseNet regresija - gubitak kroz epohe



7. DenseNet klasifikacija - točnost kroz epohe



8. DenseNet klasifikacija - gubitak kroz epohe

Slika 5.1. Točnosti i gubitci svih modela prilikom faze učenja

Radi lakšeg pregleda rezultata, u nastavku je prikazana tablica s točnošću modela na neviđenim podacima, tj. točnošću u najboljoj epohi na skupu za provjeru te točnošću na skupu za testiranje.

Model	Točnost na skupu za provjeru	Točnost na skupu za testiranje
ResNet - regresija	92.29%	56.08%
ResNet - klasifikacija	68.23%	39.81%
DenseNet - regresija	82.89%	50.55%
DenseNet - klasifikacija	58.90%	40.28%

Tablica 5.1. Točnosti modela na neviđenim slikama

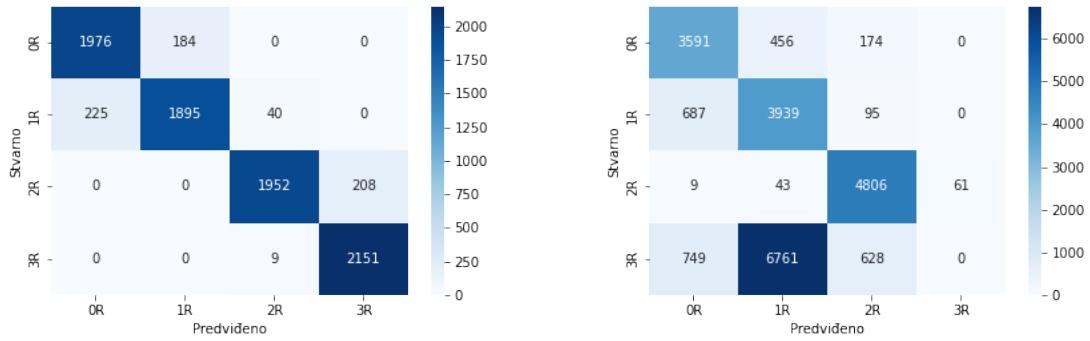
5.2. Matrice zabune

Iako točnost i gubitak pružaju korisne informacije o učinkovitosti modela, oni ne nude potpunu sliku. Zbog toga se koristi matrica zabune koja pruža detaljniji uvid u to kako model klasificira primjere unutar svake klase. Matrica zabune je tablica koja prikazuje broj točnih i netočnih predikcija modela za svaku klasu. Redovi predstavljaju stvarne klase, a stupci predviđene klase, što omogućuje analizu modelove sposobnosti da ispravno prepozna svaku klasu i identificira pogrešno klasificirane primjere.

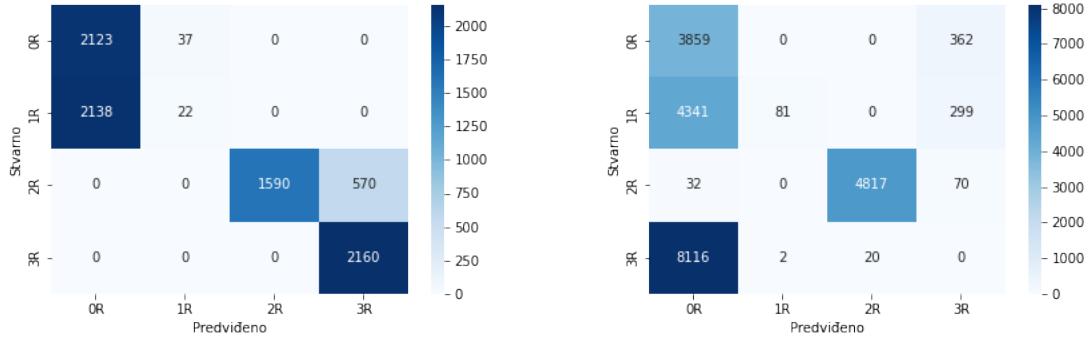
Na slici 5.2. prikazane su matrice zabune za četiri modela, kako na skupu za provjeru, tako i na skupu za testiranje. Usporedba matrica između dvaju skupova pokazuje očekivane rezultate – modeli postižu bolje rezultate na skupu za provjeru u odnosu na skup za testiranje. Razlog tome je razlika u kontrastu slika iz skupa za učenje/provjeru i slika iz skupa za testiranje prikazanog na slici 4.1. Miješanje klasa "0R" i "1R", te "2R" i "3R" također je očekivano, jer čak i među stručnjacima često postoje različita mišljenja o tome pripada li primjer stupnju odbacivanja "0R" ili "1R", odnosno "2R" ili "3R". Međutim, primjetno je da svi modeli pokazuju potpuno pogrešno klasificiranje primjera za klasu "3R" na skupu za testiranje. Ovo je posebno zabrinjavajuće jer može rezultirati "false negative" klasifikacijama, što nije poželjno.

Regresijski modeli, bilo da se radi o ResNet-u ili DenseNet-u, pokazuju konzistentne rezultate s većinom točno klasificiranih primjera unutar svake klase, osim u slučaju klase

"3R" na skupu za testiranje. S druge strane, klasifikacijski modeli imaju veći broj pogrešaka, ali pokazuju veću sigurnost u klasifikacijama, bilo one ispravne ili pogrešne. To može ukazivati na njihove poteškoće u generalizaciji na nepoznate podatke, što se posebno vidi na skupu za testiranje.



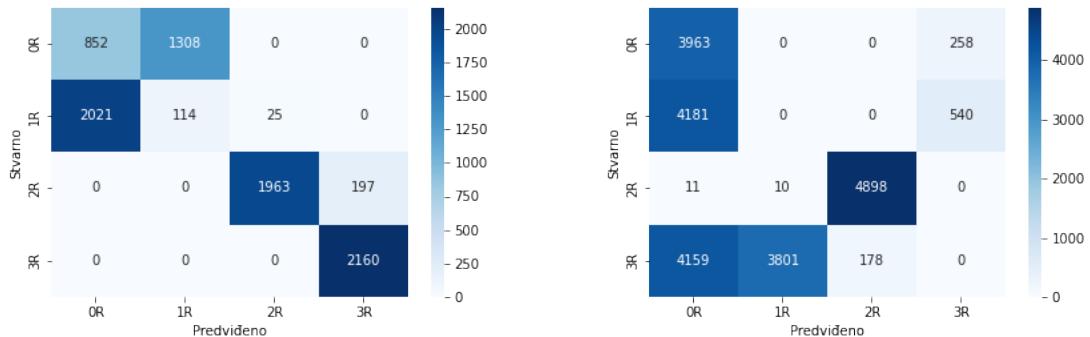
1. ResNet regresija - matrica zabune na skupu za provjeru **2. ResNet regresija - matrica zabune na skupu za testiranje**



3. ResNet klasifikacija - matrica zabune na skupu za provjeru **4. ResNet klasifikacija - matrica zabune na skupu za testiranje**



5. DenseNet regresija - matrica zabune na skupu za provjeru **6. DenseNet regresija - matrica zabune na skupu za testiranje**



7. DenseNet klasifikacija - matrica zabune na skupu za provjeru **8. DenseNet klasifikacija - matrica zabune na skupu za testiranje**

Slika 5.2. Matrice zabune svih modela na neviđenim slikama

5.3. Područja aktivacije

Osim numeričkih rezultata, vizualizacija rezultata može biti ključna za bolje razumijevanje i interpretaciju modela. Vizualni prikazi omogućuju dublji uvid u ponašanje modela, čineći ga transparentnijim i lakšim za tumačenje, što je posebno važno kada modeli dubokog učenja donose odluke na temelju složenih medicinskih slika. U zadacima analize slika, poput ovog, vizualizacije područja aktivacije neuronskih mreža igraju ključnu ulogu jer pomažu u otkrivanju regija na slici koje su najviše utjecale na odluku modela.

U ovom radu, prikaz područja aktivacije ostvareno je korištenjem tehnike Grad-CAM (*Gradient-weighted Class Activation Mapping*). Grad-CAM je napredna tehnika koja generira toplinske mape, omogućujući uvid u to koja su područja slike najviše doprinijela donošenju odluke modela. Na taj način Grad-CAM ne samo da vizualizira ključne dijelove slike već pruža i mogućnost provjere korespondencije između modelove odluke i relevantnih medicinskih značajki. Pritom se osigurava da model ne donosi odluke na temelju irelevantnih ili nevažnih značajki, što je ključno u osjetljivim područjima kao što su dijagnoze bolesti na temelju medicinskih slika.

Grad-CAM funkcioniра na principu analize gradijenata koji teku kroz konvolucijske slojeve mreže. Ovi gradijenti omogućuju identificiranje područja koja su bila ključna za aktiviranje određenih neurona u mreži, što na kraju rezultira donošenjem odluke o klasifikaciji ili predikciji. Korištenjem gradijenata vezanih uz određenu klasu, Grad-CAM generira kartu aktivacije koja prikazuje koja područja slike najviše doprinose toj klasi.

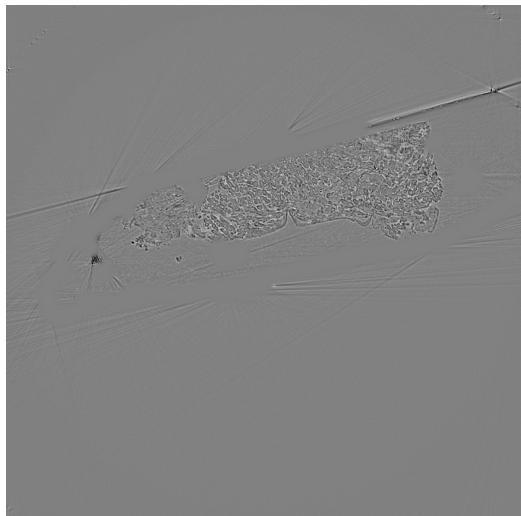
Budući da se u potpoglavlјima 5.1. i 5.2. pokazalo da se rezultati između arhitektura ne razlikuju značajno, za prikaz područja aktivacije na slici 5.3. odabrana su područja aktivacije dvaju ResNet modela. Razlog tome leži u činjenici da su razlike između regresijskih i klasifikacijskih modela znatnije, što opravdava detaljniju analizu upravo tih modela. Da bi se dobio što jasniji i pregledniji prikaz, analizirane su samo dvije slike koje prikazuju najmanji i najveći stupanj odbacivanja presatka – stupnjeve "0R" i "3R". Uzete su slike iz skupa za provjeru, koje su kontrastnije, čime su i čitatelju i neuronskoj mreži omogućeni bolji uvjeti za detekciju stanične infiltracije ili njezina izostanka. Osim toga, slike koje su odabrane bile su točno klasificirane. Područja aktivacije na skupu za testiranje pokazuju slične obrasce, iako su aktivacije nešto slabije, što je očekivano s obzirom

na razlike u kvaliteti slika između skupa za provjeru i skupa za testiranje. Ova razlika može se pripisati varijacijama u kontrastu između skupa za učenje i testnog skupa, pri čemu su testne slike često manje jasne.

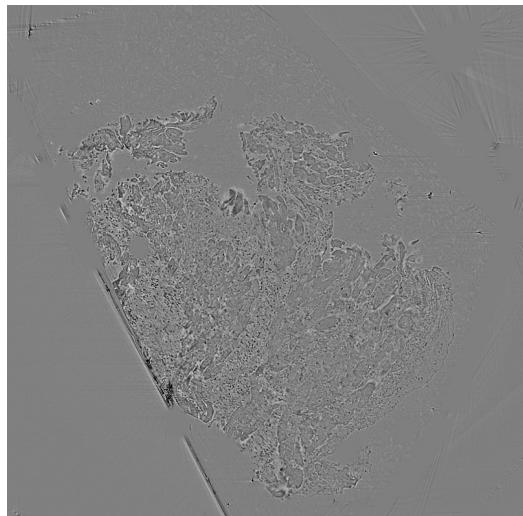
Prilikom usporedbe područja aktivacije dvaju modela, odmah je uočljivo da klasifikacijski model pokazuje više pravokutne blokove aktivacije, dok regresijski model prikazuje aktivacije s neravnim, prirodnijim obrisima, što upućuje na preciznije ciljanje relevantnih područja slike. Ovaj kontrast između pravokutnih i organskih oblika aktivacije implicira da regresijski model bolje prepozna ključna područja slike u odnosu na klasifikacijski model. Još je zanimljivije to što dva modela fokusiraju aktivacije na potpuno različita područja presatka, neovisno o tome radi li se o stupnju odbacivanja "0R" ili "3R". Pregledom stvarnih slika (bez Grad-CAM-a) može se uočiti da regresijski model uspješnije identificira točna područja presatka, što je potvrđeno činjenicom da se aktivacije pojavljuju upravo na onim dijelovima slike gdje su prisutni stanični infiltrati.

Također, iznenadujuće je da razlika u jačini aktivacija između stupnjeva "0R" i "3R" kod klasifikacijskog modela pokazuje veći kontrast nego kod regresijskog modela, iako bi se očekivalo da regresijski model bolje razlikuje različite stupnjeve odbacivanja zbog svoje prirode kontinuiranog učenja. Ovaj paradoks mogao bi ukazivati na to da regresijski model generalizira obrasce na drugačiji način od klasifikacijskog, fokusirajući se na suptilnije, ali važnije značajke, dok klasifikacijski model može pojačati kontraste između klasa čak i kad oni nisu suštinski relevantni za zadatku.

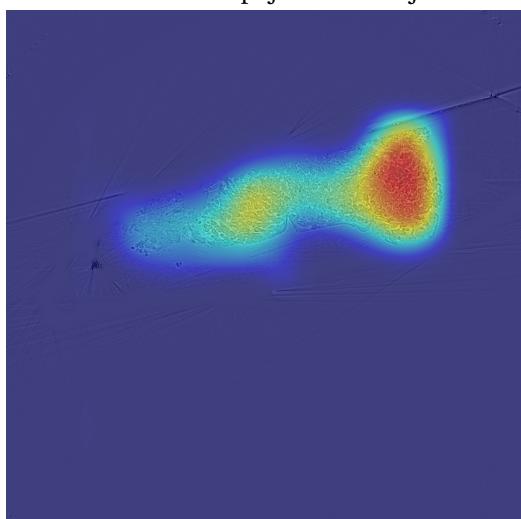
Na kraju, potrebno je objasniti zašto su područja aktivacije relativno mutna i zašto ne prikazuju finije detalje slike. To je zbog toga što su ResNet i DenseNet vrlo duboke neuronske mreže koje apstrahiraju složenje obrasce kroz mnoge slojeve. Kako se ide dublje u mrežu, receptivno polje (područje slike koje aktivira određeni neuron) postaje sve veće, što dovodi do šire i manje precizne aktivacije. Umjesto da se fokusiraju na specifične lokalne značajke, dublje mreže imaju tendenciju identificirati šire, globalne obrasce. Ovakav izgled područja aktivacije sugerira da priroda ovog problema možda zahtijeva plića neuronske mreže koje bi mogle bolje lokalizirati ključne značajke na slici. Pliće mreže često imaju manje receptivno polje, što omogućuje precizniju lokalizaciju i identificiranje specifičnih područja slike koja su od ključnog značaja za dijagnozu kod medicinskih slika.



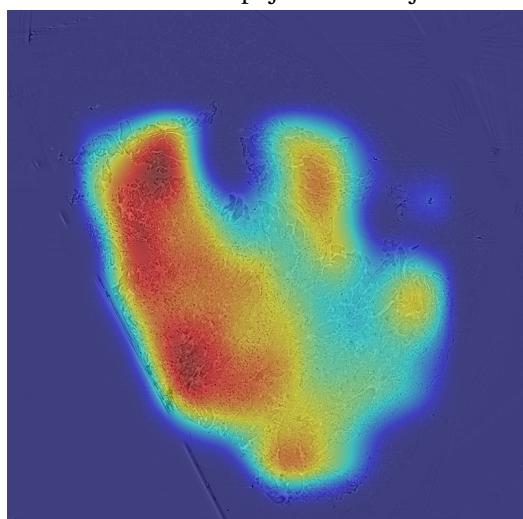
1. X-PCI slika stupnja odbacivanja "0R"



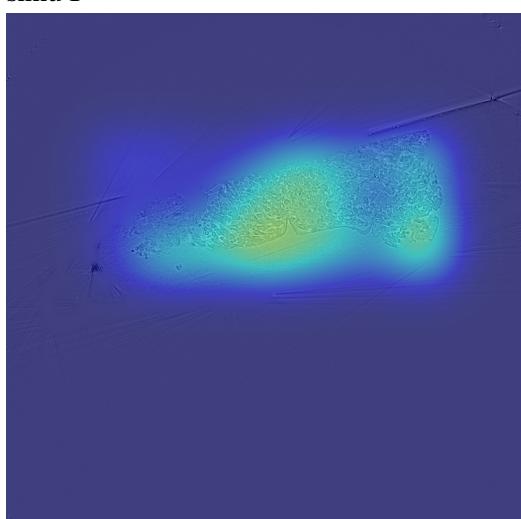
2. X-PCI slika stupnja odbacivanja "3R"



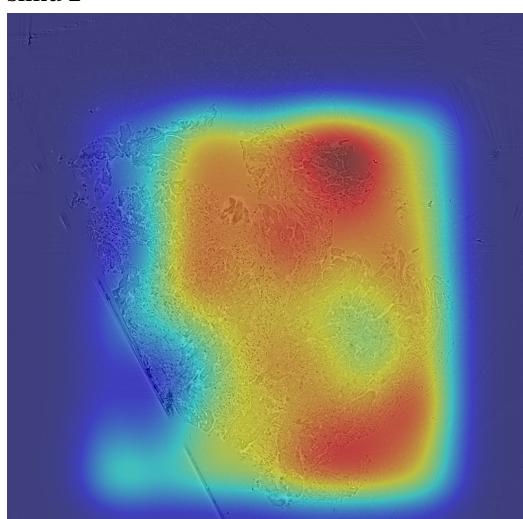
3. ResNet regresija - područja aktivacije za sliku 1



4. ResNet regresija - područja aktivacije za sliku 2



5. ResNet klasifikacija - područja aktivacije za sliku 1



6. ResNet klasifikacija - područja aktivacije za sliku 2

Slika 5.3. Područja aktivacije ResNet modela za stupanj odbacivanja "0R" i "3R" iz skupa za provjeru

5.4. Interpretacija rezultata i ograničenja

Analizom dobivenih rezultata dolazimo do zaključka da su regresijski modeli pokazali bolje performanse od klasifikacijskih modela. To je u skladu s prirodom problema, koji je u osnovi regresijski, budući da su klase gradirane, tj. imaju redoslijednu strukturu. Zbog te gradacije, regresijski modeli bolje aproksimiraju stupnjeve odbacivanja, što je ključan element u našem zadatku. Također, rezultati pokazuju da se performanse arhitektura neuronskih mreža, konkretno ResNet i DenseNet, ne razlikuju značajno, iako se pliča ResNet arhitektura pokazala nešto boljom u nekim slučajevima. Ovo može ukazivati na to da kompleksnost dubljih modela nije nužno prednost u ovakvim problemima, osobito kada je veličina skupa podataka ograničena.

Osim toga, važno je naglasiti nekoliko problema koji su se pojavili tijekom eksperimenta. Premda korištenje predtreniranih modela donosi brojne prednosti, kao što su bolja inicijalizacija kroz detekciju osnovnih značajki poput rubova, oblika i tekstura te smanjenje rizika od pretreniranosti ili podtreniranosti, važno je prepoznati i određena ograničenja. Jedno od ključnih ograničenja jest da predtrenirani modeli dolaze s općenitim značajkama naučenim na svakodnevnim slikama iz nespecijaliziranih datasetova poput ImageNet-a. Takvi modeli nisu nužno dobro prilagođeni za specifične zadatke poput analize medicinskih slika, gdje su potrebne mnogo finije značajke. Zbog toga je potrebno provesti dodatno treniranje i prilagodbu kako bi modeli mogli učinkovito naučiti specifične karakteristike medicinskih slika, što zahtijeva više vremena i resursa.

Također, postoji značajan izazov u vezi s razlikama u kvaliteti slika između različitih skupova podataka. Naime, slike u skupovima za učenje i provjeru bile su više kontrastne, što je omogućilo modelima lakše prepoznavanje ključnih značajki, poput rubova, tekstura i oblika, potrebnih za donošenje točnih predikcija. Međutim, slike u skupu za testiranje bile su znatno nižeg kontrasta, što je otežalo modelima pravilno generaliziranje. Ova razlika između kontrasta slika mogla je negativno utjecati na performanse modela na skupu za testiranje, s obzirom na to da su trenirani i validirani na slikama koje su bile jasnije i lakše za interpretaciju. Takve razlike između skupova podataka predstavljaju važan problem koji treba adresirati kako bi modeli bili bolje prilagođeni stvarnim kliničkim uvjetima, gdje slike često variraju u kvaliteti i uvjetima snimanja.

6. Zaključak

U ovom radu istražena je primjena dubokih neuronskih mreža, konkretno modela DenseNet i ResNet, u analizi medicinskih slika biopsija transplantiranih srca. Cilj je bio evaluirati učinkovitost ovih modela u klasifikaciji i regresiji stupnjeva odbacivanja presatka.

Rezultati su pokazali da oba regresijska modela, DenseNet i ResNet, postižu visoku točnost u klasifikaciji slika biopsija, posebice na visokokvalitetnim preparatima, s time da je ResNet pokazao nešto bolje performanse u usporedbi s DenseNet modelom. S druge strane, klasifikacijski modeli su pokazivali slabije rezultate zbog prirode samog problema, koji je u osnovi regresijski.

Iako su modeli pokazali obećavajuće rezultate, s trenutnom razinom preciznosti i generalizacije oni još nisu spremni za kliničku upotrebu. Glavni izazovi uključuju njihovu ovisnost o visokokvalitetnim slikama i nepredvidivo ponašanje u odnosu na slike niže kvalitete, što je čest slučaj u stvarnim kliničkim uvjetima. Nadalje, postoji problem s preciznošću detekcije u najvišim stupnjevima odbacivanja, poput "3R", što može imati ozbiljne posljedice u kliničkoj praksi. Da bi modeli bili primjenjivi u stvarnim kliničkim uvjetima, potrebno je provesti dodatne prilagodbe kojima bi se bavila buduća istraživanja.

Buduća istraživanja trebala bi se fokusirati na rješavanje ovih problema i daljnje unapređivanje modela. Prije svega, trebalo bi se usmjeriti na razvijanje regresijskih modela koji bolje odgovaraju prirodi problema. Nadalje, istraživanje bi moglo uključivati trening pličih modela, bilo da su predtrenirani ili trenirani od nule, kako bi se smanjila kompleksnost i potencijalna pretreniranost. Razvoj vlastitog modela specifičnog za medicinske slike mogao bi biti također produktivan smjer. Osim toga, moglo bi biti korisno eks-

perimentirati s treniranjem modela na slikama nižeg kontrasta kako bi se model bolje prilagodio varijacijama u slikama koje su suptilnije. Konačno, daljnja istraživanja trebala bi se usmjeriti na pronalaženje anomalija unutar skupa slika koje prikazuju jedno srce, a koje su klasificirane u isti stupanj odbacivanja, poput "3R". Primjerice, od 2000 slika koje pripadaju jednom srcu, samo manji broj slika, npr. 200, može zaista prikazivati stupanj "3R", dok ostatak može prikazivati ranije stupnjeve odbacivanja, poput "0R" ili "1R". Ova varijabilnost unutar jednog srca odražava stvarnu kliničku situaciju, gdje progresija bolesti može varirati unutar različitih dijelova istog organa. Razumijevanje tih anomalija moglo bi značajno poboljšati performanse modela i njihovu sposobnost točne klasifikacije u stvarnim kliničkim uvjetima.

Zaključno, ovaj rad pokazao je potencijal primjene dubokog učenja u analizi medicinskih slika, ali je i identificirao ključne izazove koje je potrebno prevladati kako bi se postigla potpuna primjenjivost ovih metoda u stvarnom kliničkom okruženju. Daljnje istraživanje usmjereno na rješavanje navedenih problema moglo bi značajno unaprijediti ovu tehnologiju i otvoriti vrata njezinoj široj upotrebi u medicinskoj dijagnostici.

Literatura

- [1] B. Dalbelo Bašić, M. Čupić, i J. Šnajder, *Umjetne neuronske mreže*. Zagreb: Zavod za elektroniku, mikroelektroniku i inteligentne sustave, Fakultet elektrotehnike i računarstva, Sveučilište u Zagrebu, May 2008.
- [2] ——, *Umjetne Neuronske mreže*, 2019., akademska godina 2019./2020. [Mrežno]. Adresa: [https://www.fer.unizg.hr/_download/repository/UI_12_UmjetneNeuronskeMreze\[1\].pdf](https://www.fer.unizg.hr/_download/repository/UI_12_UmjetneNeuronskeMreze[1].pdf)
- [3] M. Cheung, J. Shi, O. Wright, X. Liu, L. Jiang, i J. Moura, “Graph signal processing and deep learning: Convolution, pooling, and topology”, 07 2020.
- [4] F. Ramzan, M. U. Khan, A. Rehmat, S. Iqbal, T. Saba, A. Rehman, i Z. Mehmood, “A deep learning approach for automated diagnosis and multi-class classification of alzheimer’s disease stages using resting-state fmri and residual neural networks”, *Journal of Medical Systems*, sv. 44, 12 2019. <https://doi.org/10.1007/s10916-019-1475-2>
- [5] C. Raju, Ashoka, i B. Prakash, “Hybridtransfernet: Advancing soil image classification through comprehensive evaluation of hybrid transfer learning”, 06 2023. <https://doi.org/10.21203/rs.3.rs-3032907/v1>
- [6] I. Planinc, I. Ilic, H. Dejea, P. Garcia-Canadilla, H. Gasparovic, H. Jurin, D. Milicic, B. Skoric, M. Stampaoni, B. Bijnens, A. Bonnin, i M. Cikes, “A novel three-dimensional approach towards evaluating endomyocardial biopsies for follow-up after heart transplantation: X-ray phase contrast imaging and its agreement with classical histopathology”, *Transplant International*, sv. 36, str. 11046, Jan 2023., published 2023 Jan 24. <https://doi.org/10.3389/ti.2023.11046>

- [7] A. M. From, J. J. Maleszewski, i C. S. Rihal, "Current status of endomyocardial biopsy", *Mayo Clinic Proceedings*, sv. 86, br. 11, str. 1095–1102, Nov 2011., review.
<https://doi.org/10.4065/mcp.2011.0296>

Sažetak

Analiza kardioloških slika korištenjem duboke neuronske mreže

Adam Vuković

U ovom radu istražena je primjena dubokih neuronskih mreža za analizu kardioloških slika biopsija transplantiranih srca. Rad započinje s objašnjavanjem osnova neuronskih mreža, a zatim se bavi implementacijom DenseNet i ResNet arhitektura koje su testirane na X-PCI slikama. Na samome kraju slijedi analiza rezultata te rasprava o dalnjim istraživanjima.

Ključne riječi: umjetne neuronske mreže; konvolucijske neuronske mreže; X-PCI; biopsije transplatičnih srca; PyTorch; DenseNet; ResNet

Abstract

Cardiac Image Analysis Using Deep Neural Network

Adam Vuković

This paper seeks to investigate the utilization of deep neural networks in the analysis of cardiac images obtained from transplanted heart biopsies. The initial section of the paper provides an overview of artificial neural networks, followed by a detailed explanation of the implementation of DenseNet and ResNet architectures, both of which are evaluated using X-PCI images. The paper concludes with an analysis of the findings and a discussion on potential future research directions.

Keywords: artificial neural networks; convolutional neural networks; X-PCI; transplanted hearts biopsies; PyTorch; DenseNet; ResNet