

Predtreniranje dubokih neuronskih mreža na analizu medicinskih slika

Cvitanović, Petar

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:217654>

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

Download date / Datum preuzimanja: **2025-03-28**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 415

**PREDTRENIRANJE DUBOKIH NEURONSKIH MREŽA NA
ANALIZU MEDICINSKIH SLIKA**

Petar Cvitanović

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 415

**PREDTRENIRANJE DUBOKIH NEURONSKIH MREŽA NA
ANALIZU MEDICINSKIH SLIKA**

Petar Cvitanović

Zagreb, lipanj 2024.

DIPLOMSKI ZADATAK br. 415

Pristupnik: **Petar Cvitanović (0036518121)**
Studij: Računarstvo
Profil: Računarska znanost
Mentor: akademik prof. dr. sc. Sven Lončarić

Zadatak: **Predtreniranje dubokih neuronskih mreža na analizu medicinskih slika**

Opis zadatka:

Duboko učenje važno je u obradi medicinskih slika jer ubrzava i poboljšava kliničku dijagnostiku. Jedna od primjena je klasifikacija stupnja odbacivanja srčanog tkiva kod pacijenata s transplantiranim srcem. No, za razliku od nekih drugih područja računalnog vida, anotiranje medicinskih slika je dugotrajan i skup proces jer zahtjeva ekspertno znanje, te su zato metode za predtreniranje od iznimne važnosti u ovom području. U okviru diplomskog rada, potrebno je proučiti postojeće metode za predtreniranje dubokih neuronskih mreža, s naglaskom na metode koje se koriste kod transformatorskih arhitektura. Potrebno je odabrati jednu od proučenih metoda, implementirati ju te ju iskoristiti za predtreniranje duboke neuronske mreže. Točnost implementirane mreže potrebno je provjeriti na ispitnom skupu slika te usporediti rezultate s drugim metodama iz literature.

Rok za predaju rada: 28. lipnja 2024.

*Zahvaljujem se prof. dr. sc. Svenu Lončariću na mentorstvu i Doniku Vršnaku, mag. ing.,
na pomoći u tehničkoj implementaciji i na dostupnosti za sva pitanja.*

Sadržaj

1. Uvod	3
1.1. Cilj	3
2. Strojno učenje	5
2.1. Neuronske mreže	5
2.2. Multiple instance learning	6
2.2.1. Primjene	7
3. Implementacija	9
3.1. Skup podataka	9
3.1.1. Poznati šumovi u skupu podataka	14
3.2. Neuronska mreža	14
3.3. Multiple instance learning	16
4. Rezlutati	18
4.1. Neuronska mreža sa 4 klase	18
4.2. Neuronska mreža sa 2 klase	20
4.3. Multiple instance learning model	21
5. Moguća poboljšanja	25
5.1. Prilagodba modela	25
5.2. Prilagodba skupa podataka	26
6. Zaključak	27
Literatura	29

Sažetak	31
Abstract	32

1. Uvod

Danas su razvijene i testirane mnoge neinvazivne metode dijagnosticiranja akutnog staničnog odbacivanja transplantata nakon transplantacije srca. Pri dijagnozi akutnog odbacivanja često se promatra kako transplantant funkcionira ili se detektiraju imunološki procesi koji indiciraju odbacivanje transplantata. Uspješnost takvih metoda dijagnosticiranja uspoređuje se s analizom uzoraka tkiva srca dobivenih biopsijom zbog dobivanja što točnijih rezultata.

Histološkom analizom, odbacivanje srca nakon transplantacije, vidljivo je kao nakupljanje bijelih krvnih stanica, a u težim slučajevima vidljiva je i šteta na tkivu prouzrokovana prodorom bijelih krvnih stanica.

Kod blažih odbacivanja, vidljivi su samo infiltrati bijelih krvnih stanica s ponekim oštećenjima strukture srčanog tkiva, no kod ozbiljnijih slučajeva odbacivanja, sve veće površine su zahvaćene oštećenjima, koje mogu rezultirati krvarenjem.

Radi lakšeg rangiranja odbacivanja transplantanata, 2005. godine je Međunarodno društvo za transplantaciju srca i pluća objavilo sustav ocjenjivanja odbacivanja transplantanata. Sustav se koristi u većini transplantacijskih centara diljem svijeta. Što je više infiltrata i oštećenja na tkivu, veći je stupanj odbacivanja. Sustav ocjenjivanja i razine odbacivanja detaljnije su opisane u poglavlju 3.1.

1.1. Cilj

Cilj rada je istražiti i implementirati model strojnog učenja sposoban za klasifikaciju slika s histološke analize u spomenute razine. Za treniranje i validaciju modela, dostupan je skup podataka koji se sadrži od slika srca na kojima su potencijalno vidljivi tragovi odbacivanja.

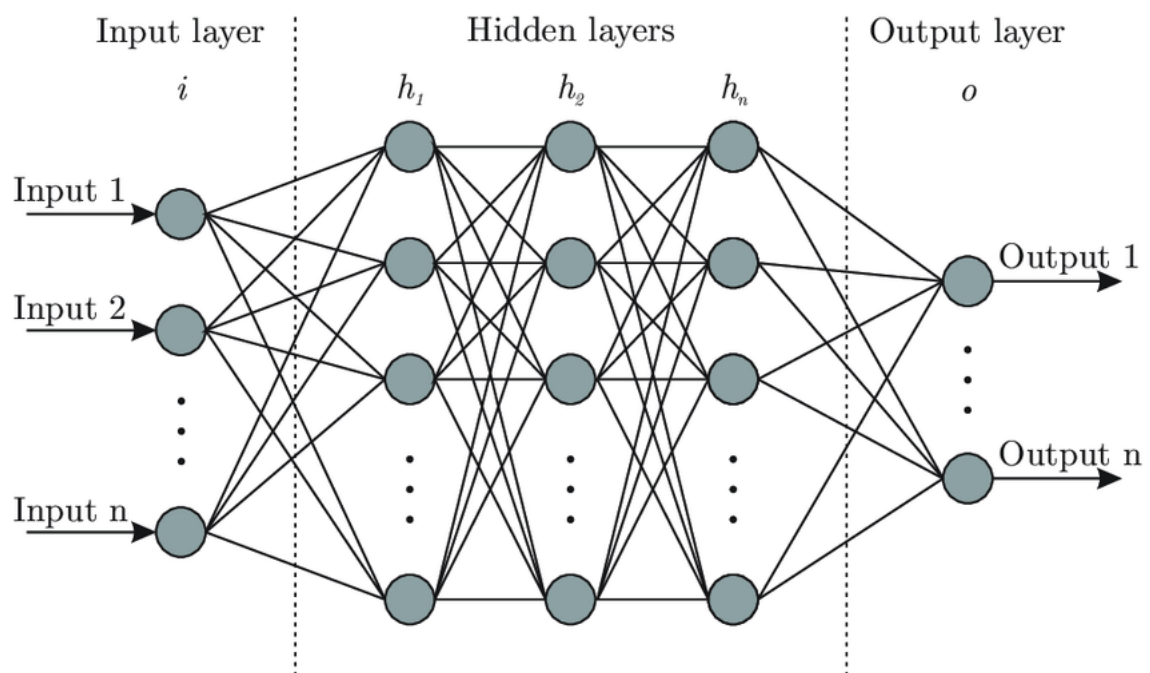
Slike iz skupa podataka su podijeljene na manje slike, te je za treniranje modela korištena metoda multiple instance learning kojom će se kroz rad proširiti model koji koristi neuronsku mrežu. Usporedit će se obična neuronska mreža s onom proširenom multiple instance learning metodom na temelju preciznosti i brzini konvergencije.

2. Strojno učenje

Strojno učenje danas ima mnogo primjena i može automatizirati i olakšati svakodnevne zadatke i poslove. Tako je i pri problemu prepoznavanja odbacivanja na slikama srca odabrano strojno učenje kako bi se olakšao posao doktora. Za problem prepoznavanja odbacivanja, trenirana su dva modela strojnog učenja. Prvi od dva je neuronska mreža dok je drugi neuronska mreža prilagođena za *multiple instance learning*. Multiple instance learning je odabran kao moguće rješenje s obzirom na dostupan skup podataka koji će biti dublje opisan u kasnijim poglavljima.

2.1. Neuronske mreže

Umjetna neuronska mreža je masivno paralelni distribuirani procesor koji je dobar za pamćenje iskustvenog znanja [1].



Slika 2.1. Neuronska mreža¹

Svaka umjetna neuronska mreža sastoji se od slojeva umjetnih neurona. Slojevi neurona mogu se međusobno povezivati kako bi dobili drugačija svojstva mreže i koristili mrežu u drugačije svrhe. Zadatak mreže je naučiti i modelirati stvarni svijet. Kako bi to postigli za opisani problem, mrežu treniramo podacima koji dolaze u parovima - slika i razina odbacivanja transplantiranog srca. Na temelju danog skupa podataka neuronska mreža prilagođava svoje neurone kako bi što bolje opisala postojeći skup podataka, ali i mogla opisati nove podatke - u ovom slučaju slike srca.

2.2. Multiple instance learning

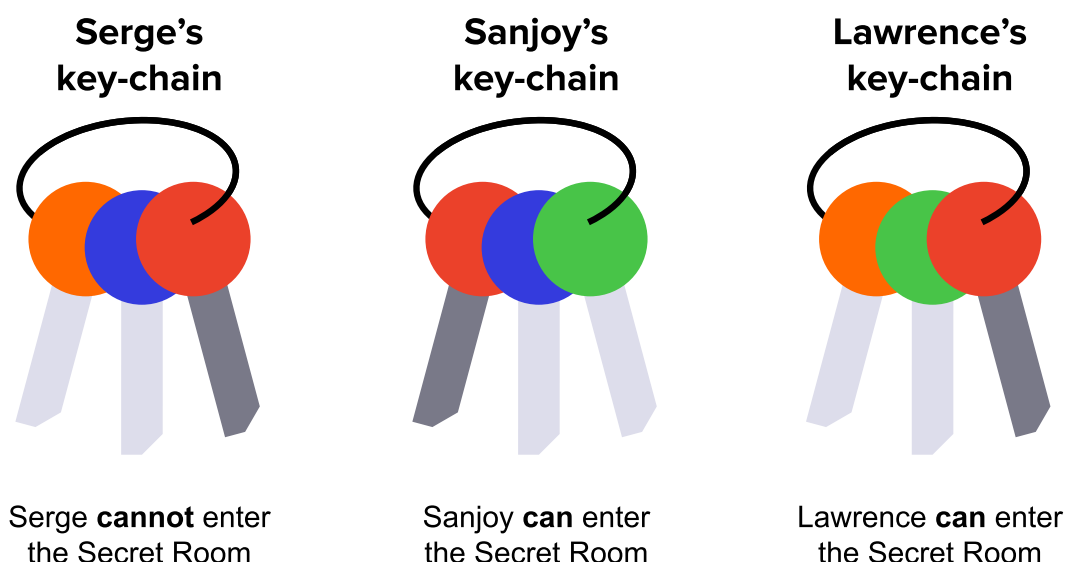
Multiple instance learning (MIL) je nadogradnja na koncept neuronske mreže. Multiple instance learning nam omogućava da treniramo i validiramo model na skupovima podataka koji se sastoje od parova u kojima imamo jednu oznaku za skupinu slika umjesto jednu sliku kao u slučaju uobičajene neuronske mreže.

Takve skupine slika koje dijele istu oznaku nazivamo vreća (en. *bag*) [3]. Svaka vreća sastoji se od više instanci podataka za treniranje ili validaciju. Ovisno o primjeni, jedna vreća može sadržavati više slika, isječak iz videa ili nekoliko povezanih tekstova, ali cijela vreća je uvijek označena samo jednom oznakom. S obzirom na to da jedna vreća sadrži više instanci za učenje, nisu sve instance bitne za konačnu klasifikaciju vreće. Od svih instanci unutar jedne vreće, samo neke mogu biti korisne za donošenje odluke, dok su druge samo šum.

Kao primjer možemo uzeti privjeske za ključeve koji će predstavljati vreće, a ključevi na tim privjescima mogu predstavljati pojedinačne instance za treniranje.

Svaki privjesak za ključeve pripada jednoj osobi za koju želimo zaključiti može li ući u određenu sobu. Ukoliko privjesak sadrži određeni ključ, osoba može ući u sobu, u protivnom ne. Među privjescima sa slike 2.2., prvi privjesak nema odgovarajući ključ za ulazak, dok druga dva privjeska imaju. Iz tih informacija možemo saznati koji ključ ili instanca, odnosno izostanak istog, nam je bitan za klasifikaciju cijelog privjeska. Također, pomoću ovog znanja možemo klasificirati ostale privjeske za koje do sada nismo znali mogu li otključati vrata.

Multiple instance learning metoda je korisna pri problemima u kojima podaci nisu



Slika 2.2. [4]

označeni dovoljno granularno. Na primjer, uz multiple instance learning metodu možemo video klasificirati kao video u kojem se pojavljuje neki objekt umjesto da svaki kadar videa označavamo s obzirom na to pojavljuje li se objekt ili ne. Na taj način smo olakšali problem označavanja skupova podataka, što znatno pomaže većim modelima kojima je potrebno mnogo podataka. Upravo iz ovog razloga je metoda dobila na popularnosti.

2.2.1. Primjene

Postoje različite primjene multiple instance learning metode: klasifikacija, regresija, rangiranje i grupiranje [3]. S obzirom na to da je MIL moguće koristiti na drugačijim modelima i primjenama, ova metoda daje mogućnost trenirati širok skup modela na slabo označenim skupovima podataka kao što su skupovi u kojima nije označena svaka instanca, nego su labelirane vreće instanci.

Klasifikaciju je moguće provesti na razini instance i na razini vreće [3]. Ovisno o razini na kojoj se provodi klasifikacija, koristit će se drugačija funkcija gubitka. Ako provedemo MIL metodu na spomenutom primjeru ključeva [4] i pri klasifikaciji ključeva u na trećem privjesku klasificiramo sve ključeve kao ključeve koji otvaraju vrata, pogriješit ćemo na razini instance, ali na razini vreće, klasifikacija ostaje točna jer smo na privjesku već imali ključ koji otvara vrata.

Koristeći multiple instance learning nad regresijom možemo vreći pridodati vrijednost umjesto oznake. Postoje razne metode kako odabrati vrijednost cijele vreće. Neke od njih koriste prosjek ili druge težinske funkcije kako bi se temeljem instanci pridodala vrijednost svakoj vreći.

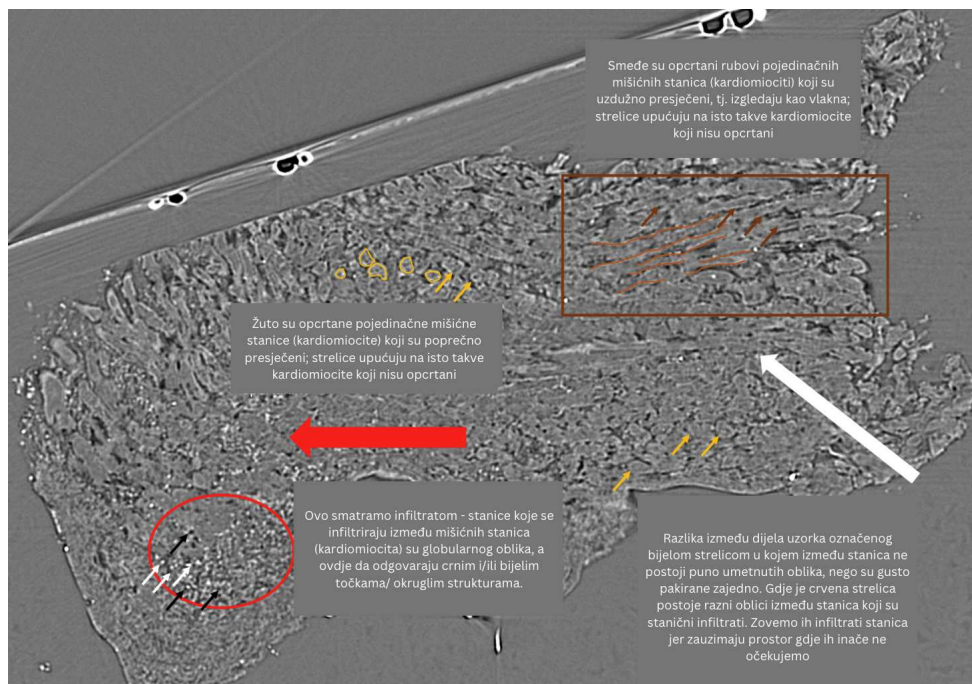
Osim spomenutih primjena, moguće je koristiti multiple instance learning prilikom rangiranja i grupiranja. Prilikom rangiranja, cilj je odrediti odnos vreća umjesto pridodati im vrijednosti. Slično, grupiranjem se postiže struktura vreća umjesto struktura instanci, pri čemu se mogu koristiti postojeće mjere udaljenosti.

3. Implementacija

U ovom poglavlju dublje će se ući u tehničke detalje rada. Objasnit će se struktura skupa podataka i dva modela trenirana za prepoznavanje odbacivanja srca na slikama. Za rješavanje problema implementirana su dva modela strojnog učenja: obična neuronska mreža i neuronska mreža proširena multiple instance learning metodom. Rješenje je implementirano u Pythonu. Za izgradnju neuronskih mreža i učitavanje skupa podataka korištena je python biblioteka PyTorch.

3.1. Skup podataka

Dostupan skup podataka sadrži slike srca označene razinom odbacivanja. Na slikama je moguće vidjeti presjeke mišićnih stanica srca koje nazivamo kardiomiocitima.



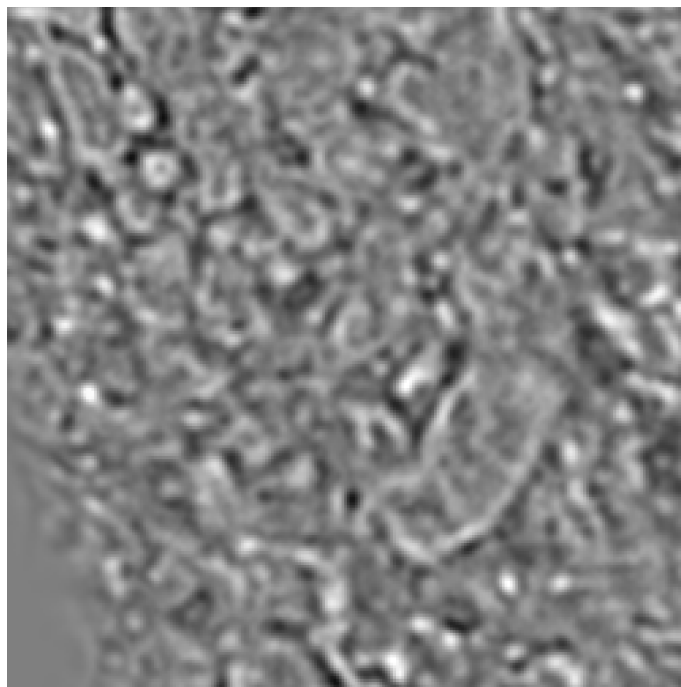
Slika 3.1. Primjer slike iz skupa podataka

Na slici 3.1. vidljiva je jedna potpuna slika iz skupa podataka. Takva slika je u skupu podataka podjeljena u dijelove kao što je objašnjeno dalje u poglavlju i predstavlja sve slike jedne vreće.

Slika prikazuje presjek srca i stanice tkiva. Pri sredini slike, neke stanice su označene žutom bojom i predstavljaju kardiomiocyte koji su na snimci presječeni poprečno te izgledaju kao krugovi na slici. Na desnom dijelu slike, smeđom bojom su prikazani kardiomiociti presječeni uzdužno, te zato na slici izgledaju kao vlakna.

Na dijelu slike koji se nalazi dolje lijevo, prikazan je primjer infiltrata, odnosno viša koncentracija bijelih krvnih stanica. Na slici se infiltrat očituje kao nakupina bijelih i crnih točaka grupiranih na jedno mjesto.

Skup podataka strukturiran je kao ugniježđeni direktorij slika. Slike i direktoriji su raspoređeni u hijerarhiji ovisno o razini odbacivanja prikaznom na slici, te ostalim parametrima koji su manje bitni za treniranje modela. Svaki ugniježđeni direktorij na najdubljoj razini sadrži više slika. Svaka od slika odgovara jednoj instanci za treniranje modela, a sve slike u jednom direktoriju na najnižoj razini predstavljaju jednu vreću instanci.



Slika 3.2. Primjer slike iz skupa podataka

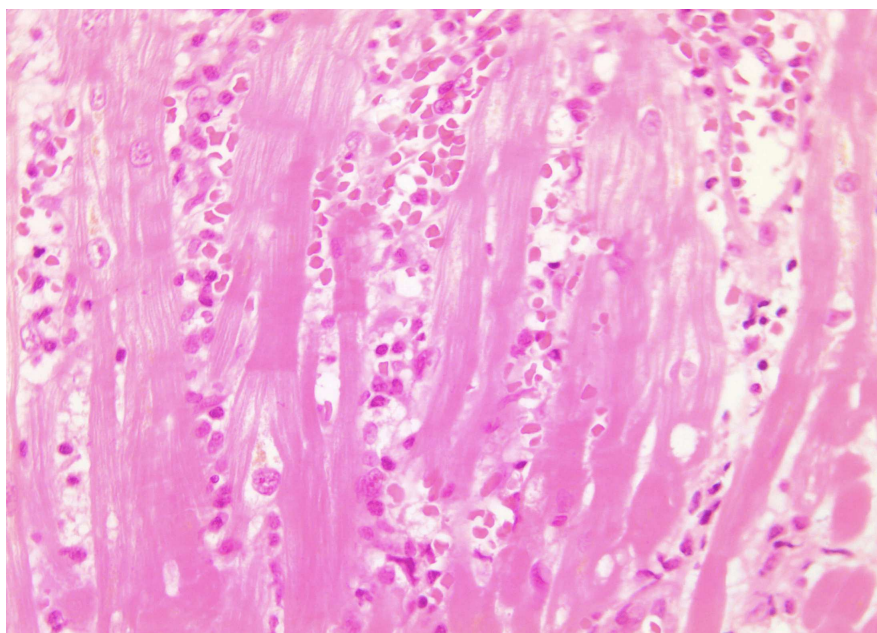
Na slici 3.2. prikazana je jedna instanca za treniranje modela. Instance nemaju za-

sebno određenu oznaku, već su oznake dodijeljene na razini vreće. Oznaka vreće definirana je u imenu direktorija svake vreće. Ime direktorija u kojem se nalazi slika 3.2. je:

/2017/08_4001_2_3R_B1/rec_8bit_08_B1/08_4001_2_3R_B10269.rec.8bit

U imenu zadnjeg ugniježđenog direktorija 08_4001_2_3R_B10269.rec.8bit stoji oznaka vreće 3R. Moguće oznake vreća su:

- 0R - Nema staničnog infiltrata te nije vidljivo akutno odbacivanje. Liječenje nije potrebno.
- 1R - Vidljivo blago akutno odbacivanje, ali liječenje nije potrebno. Odbacivanje vidljivo samo na jednom presjeku srca. Vidljivo kao područje gusto infiltratom i oštećenjem stanica.
- 2R - Umjereno stanično odbacivanje je vidljivo na slici. Karakterizira se slično kao područje s infiltratom i oštećenjem kao kod 1R, ali prisutno na više mjesta. Potrebno je liječiti.
- 3R - Vidljivo jako stanično odbacivanje koje je potrebno liječiti. Infiltrat i oštećenja su prisutni na više područja.



Slika 3.3. Primjer snimke srca sa razinom 3R

Vreće nemaju konzistentan broj instanci. Vreća iz primjera sastoji se od šest slika u .tif formatu.

Svaka vreća u skupu podataka predstavlja jednu potpunu snimku srca podijeljenu u dijelove (patcheve). Svaki patch je jedna instanca u vreći. Tako vreća iz prethodnog primjera predstavlja jednu sliku srca na kojem je prisutno odbacivanje, podijeljenu u šest dijelova.

U našem slučaju, skup podataka nije slabo označen, što je jedna od glavnih značajki zbog koje bi izabrali multiple instance learning metodu. U našem skupu podataka je svaka slika zasebno označena, ali je podijeljena u patcheve koji nisu zasebno označeni. Treniranjem modela na patchevima umjesto cijelim slikama, možemo poboljšati performance modela i smanjiti potrebne resurse za treniranje modela.

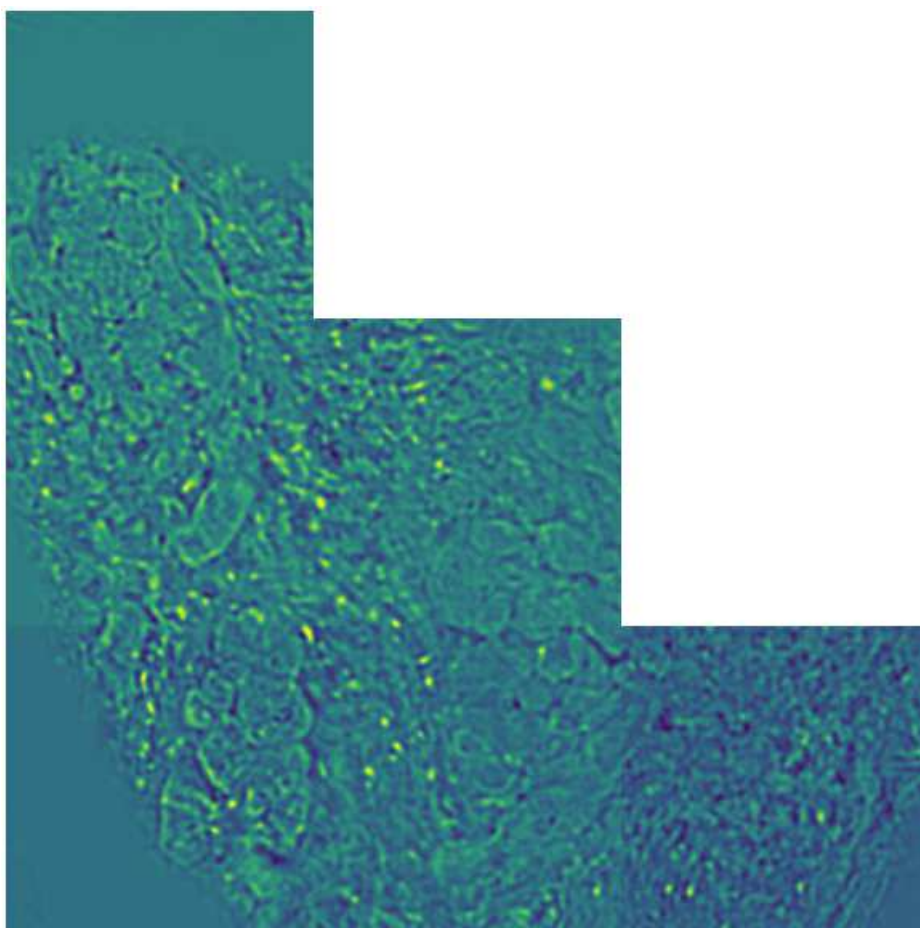
Sve instance jedne vreće možemo spojiti kako bi dobili cijelu snimku srca. Za vreću iz direktorija /2017/08_4001_2_3R_B1/rec_8bit_08_B1/08_4001_2_3R_B10269.rec.8bit, potpuna slika je prikazana na slici 3.4. ispod.

Potpuna slika 3.4. sastoji se od patcheva u vreći koji su raspoređeni na sljedeći način

```
000.tif
001.tif  002.tif
003.tif  004.tif 005.tif
```

Kako bi se model mogao trenirati na opisanim slikama, u Pythonu je implementirana skripta koja rekurzivno učitava slike i parsira oznake iz imena direktorija. Za treniranje obične neuronske mreže, svaka slika je učitana kao zasebna instanca i pridodana joj je oznaka cijele vreće. Naravno, u tom slučaju, ta oznaka ne mora biti točna jer na cijeloj slici može biti prisutno odbacivanje, ali na nekim patchevima te slike neće biti odbacivanja. Prilikom treniranja neuronske mreže proširene sa multiple instance learning metodom, slike su učitane kao vreće i oznaka je pridodana cijeloj vreći. U oba slučaja, implementirana je python klasa koja proširuje klasu Dataset iz paketa torch.utils.data.

Skup podataka sadrži 22322 slika koje su podijeljene u 1230 vreća. Distribucija slika po oznakama je prikazana ispod.



Slika 3.4. Potpuna slika snimke srca

Oznaka	Broj instanci	Postotak instanci
0R	8477	37.98%
1R	7029	31.49%
2R	3292	14.75%
3R	3524	15.79%

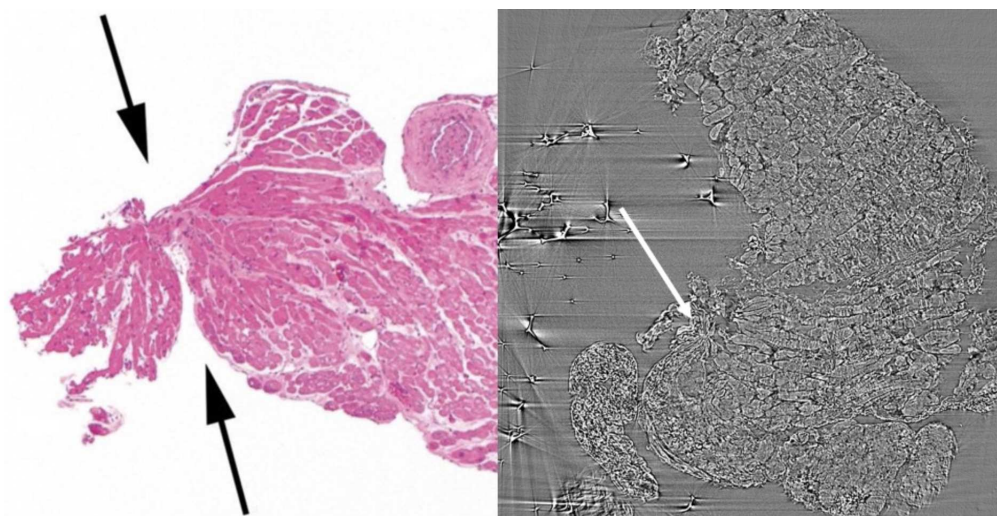
Također, distribucija vreća po oznakama:

Oznaka	Broj instanci	Postotak instanci
0R	499	40.57%
1R	318	25.85%
2R	229	18.62%
3R	184	14.96%

Kao što je prikazano u tablicama iznad, puno je više instanci i vreća za oznaka koje

označuju manju razinu odbacivanja, nego onih koji ukazuju na višu.

3.1.1. Poznati šumovi u skupu podataka



Slika 3.5. Šum izazvan biptomom

U skupu podataka, mogući su šumovi koji mogu utjecati na točnost i pouzdanost modela strojnog učenja. Jedan od čestih poznatih šumova na slikama izazvan je biptomom. Biptom je medicinski uređaj koji se koristi za endomiokardijalnu biopsiju, postupak u kojem se uzima uzorak srčanog tkiva radi daljnje analize [5]. Tijekom ovog postupka, biptom može izazvati pojavu "štibanja" tkiva, što rezultira šumovima na slikama.

Kao što je vidljivo na slici 3.5., "štibanje" tkiva uzrokovano biptomom može se manifestirati kao neobični oblici i nepravilnosti u strukturi tkiva. Ovi šumovi mogu ometati model strojnog učenja, jer dodaju složenost i varijabilnost u skup podataka koji se koristi za treniranje.

Pojava šumova uzrokovanih biptomom nije samo tehnički izazov, već može imati i kliničke implikacije. Artefakti na slikama mogu dovesti do pogrešnih dijagnoza ili netočnih zaključaka, što naglašava važnost njihove identifikacije i uklanjanja u procesu analize.

3.2. Neuronska mreža

Jedan od dva modela implementirana i trenirana za problem klasifikacije slika srca je obična neuronska mreža. Mreža je trenirana i testirana na pojedinačnim slikama kojima

su dane oznake vreća.

Kao što je spomenuto, ukoliko se na cijeloj slici, odnosno vreći može naći infiltrat koji ukazuje na odbacivanje srca, ne mora značiti da će na svakoj instanci, odnosno patchu biti vidljiv infiltrat. Iz tog razloga ćemo neke instance krivo označiti kao s višom razinom odbacivanja, iako na njima nema tragova infiltrata.

Mreža je implementirana proširivanjem klase `Module` biblioteke `torch`, te implementira konstruktor i funkciju `forward`. U konstruktoru klase, definirani su slojevi mreže i njihove dimenzije, dok funkcija `forward` prima jednu instancu iz skupa podataka za treniranje, te ga provlači kroz slojeve mreže iz kojih konačno dobiva vektor iz čijih vrijednosti možemo zaključiti klasu instance. Izlazni vektor je dužine broja klasa i što je vrijednost nekoj poziciji u vektoru veća, to je veća vjerojatnost da instanca pripada toj klasi.

Mreža se sastoji dva od konvolucijska sloja s ReLU aktivacijskim funkcijama praćena s max pool funkcijama. Nakon dva konvolucijska sloja slijede tri potpuno povezana sloja koja reduciraju dimenzije na broj mogućih klasa.

Za funkciju gubitka korišten je `CrossEntropyLoss` iz biblioteke `torch.nn`. Kao algoritam za optimizaciju parametara korišten je algoritam AdamW [6]. AdamW koristi weight decay kako bi penalizirao parametre mreže koji su preveliki. Posljedica weight decay-a je smanjivanje pretreniranosti mreže i bolju generalizaciju.

Ovako definirana neuronska mreža trenirana je na istom skupu podataka kojem su dane drugačije oznake. U prvom slučaju, mreža je trenirana na skupu koji ima četiri moguće labele 0R - 3R. U drugom slučaju, klase se grupirane u dvije klase. Klase 0R i 1R su grupirane u jednu klasu, dok su 2R i 3R grupirane u drugu klasu.

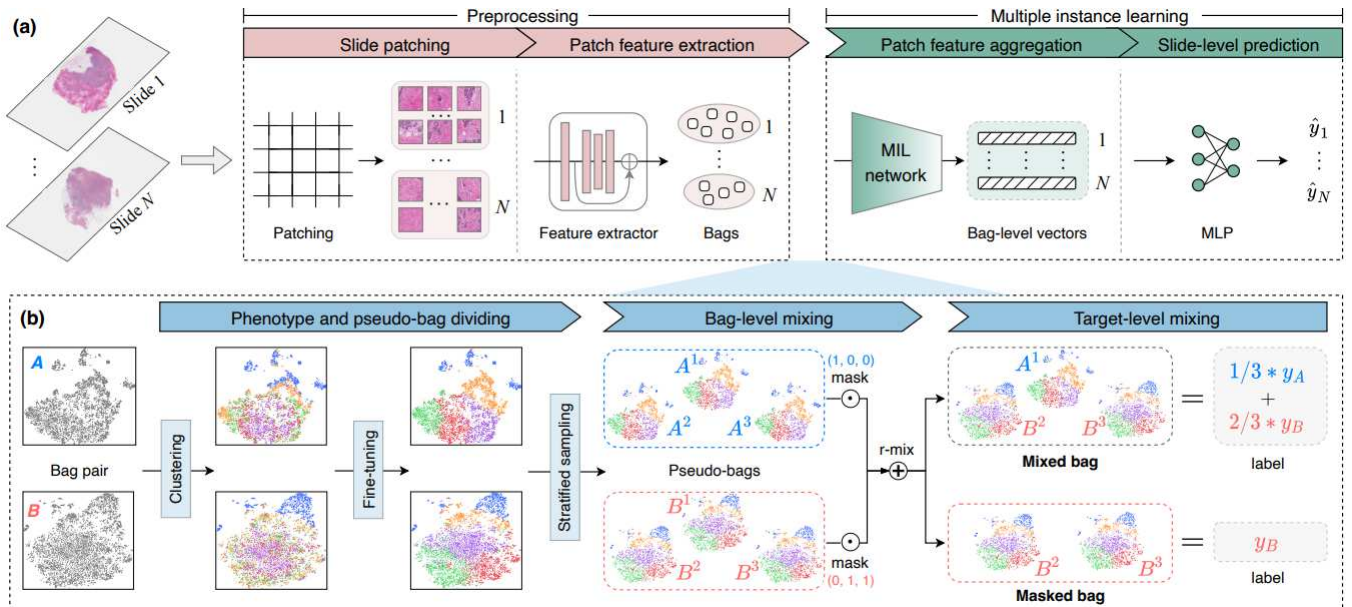
Za potrebe grupiranja, učitavanje skupa podataka je izmijenjeno kako bi se grupe ispravno grupirale, te je promijenjen zadnji sloj mreže. Zadnji sloj mreže je potpuno povezan linearni sloj čiji izlaz je vektor veličine broja klasa. Pri mijenjanju broja klasa, prilagođen je izlaz zadnjeg sloja.

3.3. Multiple instance learning

Model nadograđen multiple instance learning metodom sličan je modelu iz prošlog poglavlja. MIL model je jednak prethodnom modelu po svom izlazu na kojem predviđa klasu dane vreće u obliku vektora. Ulaz modela je drugačiji jer MIL model na svome ulazu prima vreću, odnosno niz slika, na temelju kojih predviđa klasu cijele vreće.

MIL model podijeljen je na dva dijela. Prvi dio je neuronska mreža koja je slična onoj iz prethodnog poglavlja i sastoji se od konvolucijskih slojeva s ReLU funkcijama i max pool slojeva. Drugi dio mreže agregira izlaze dobivene iz prvog dijela. S obzirom na to da prvi dio mreže ne radi s pojedinačnim slikama, već s vrećama slika, prvi dio modela koji je neuronska mreža na svom izlazu daje listu vektora u kojem su pohranjene predikcije za svaku sliku iz vreće.

Iz tog razloga, prvi dio mreže je prilagođen kako bi se niz slika mogao propuštati kroz slojeve mreže, no slojevi i njihova primjena su ostali isti. Promjena je implementirana tako da se niz slika spoji u jedan tensor koji je pušten kroz slojeve. Nakon prvog dijela, implementirana je agregacija s pomoću attention poolinga [7].



Slika 3.6. Arhitektura modela [8]

Attention pooling agregira izlaz neuronske mreže u jedan u vektor koji predstavlja

konačnu klasu cijele vreće. Postoje druge metode za agregaciju izlaza neuronske mreže, no attention pooling je najfleksibilnija s obzirom na to da je ovaj oblik agregacije građen kao neuronska mreža. U slučaju modela implementiranog u sklopu rada, attention pooling je implementiran potpuno povezanim slojevima.

Slika 3.6. prikazuje multiple instance learning model i njegovu podjelu u dva dijela. U gornjem desnom dijelu slike, prikazan je MIL network koji je prvi dio mreže i na svojem izlazu vraća vektore. Ti vektori su agregirani u jedan vektor koji predviđa oznaku cijele vreće s pomoću druge neuronske mreže prikazane u "Slide-level prediction" dijelu modela.

4. Rezlutati

Kroz ovo poglavlje navedeni su rezultati modela treniranih na opisanom skupu podataka. Za svaki model bit će prikazana funkcija gubitka, te F1 mjera kroz epohe.

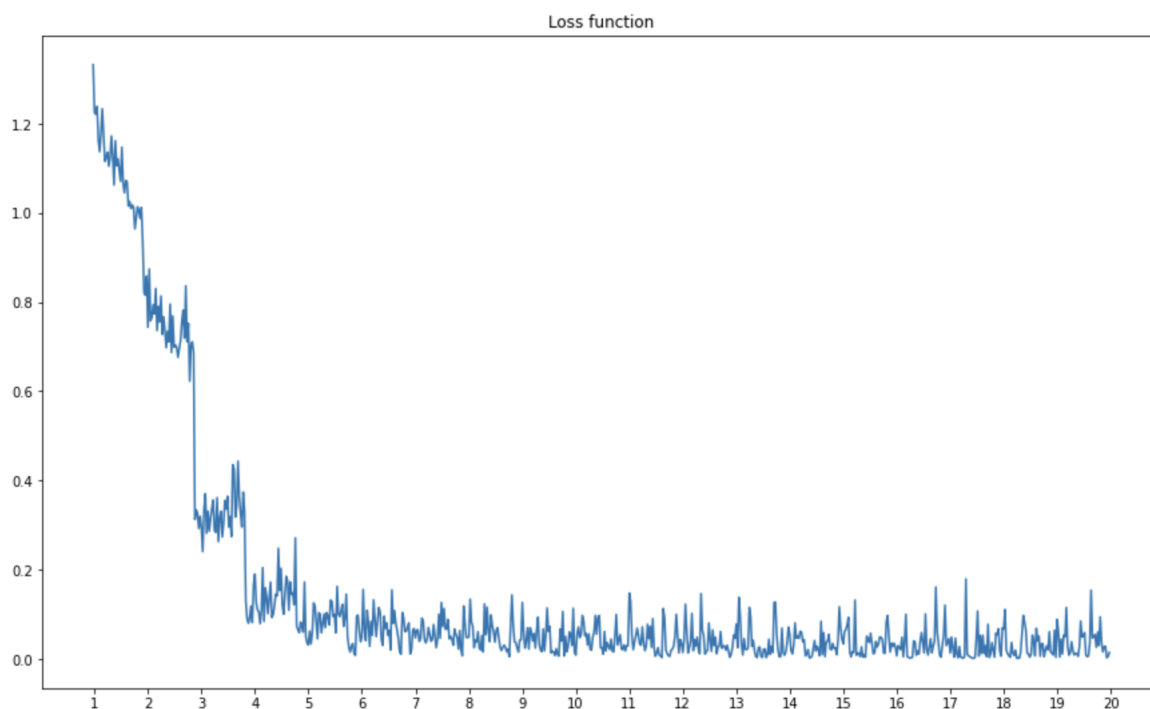
Mjera F1 se često koristi kao mjera uspješnosti predviđanja klasa kod klasifikacijskih problema. F mjera je harmonijska sredina preciznosti i odziva, gdje je preciznost udio točno klasificiranih primjera u skupu pozitivno klasificiranih primjera, a odziv udio točno klasificiranih primjera u skupu svih pozitivnih primjera [9].

Modeli su trenirani na više epoha, gdje se u svakoj epohi model trenira na cijelom skupu za treniranje, te se na kraju svake epohe računa F1 mjera što je prikazano na grafovima.

4.1. Neuronska mreža sa 4 klase

Prvi model je neuronska mreža opisana u 3.2. trenirana na skupu podataka čije su vreće podijeljene u 4 klase, odnosno klase nisu grupirane. Model je treniran na 20 epoha, te je nakon svake epohe evaluirana njegova F1 mjera na skupu za treniranje.

Na slici 4.1. prikazana je funkcija gubitka kroz treniranje mreže. Model je treniran tijekom dvadeset epoha, no primjetno je da nakon šeste epohe vrijednost funkcije gubitka prestaje padati. Početna vrijednost funkcije gubitka iznosila je 1.3324, dok je do šeste epohe smanjena na 0.05. Nakon točke u šestoj epohi, nema daljnjeg značajnog smanjenja gubitka, što ukazuje na pretreniranje modela. Pretreniranje se događa kada model postane preosjetljiv na trening podatke i gubi sposobnost generalizacije na novim, neviđenim podacima. U našem slučaju, nedostatak smanjenja funkcije gubitka nakon šeste epohe sugerira da model počinje "pamćenjem" trening skupa, umjesto učenjem općih obrazaca koji bi mu omogućili dobru performansu na novim podacima.

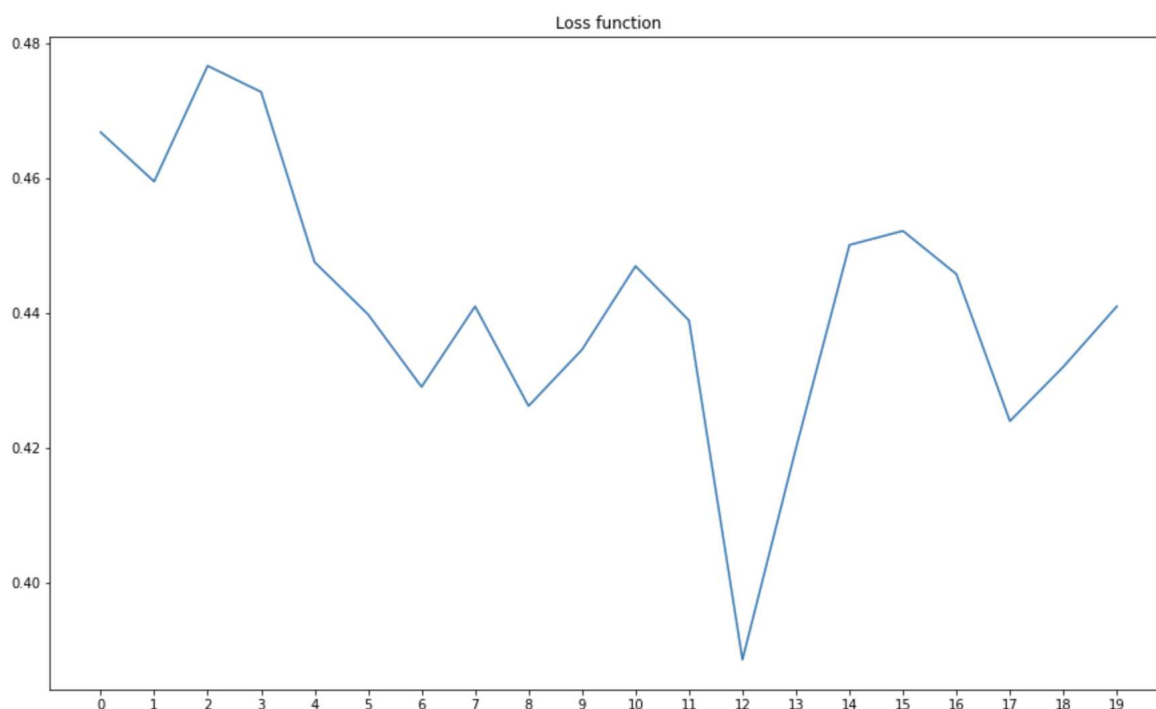


Slika 4.1. Funkcija gubitka svakih 100 iteracija

Maksimalna vrijednost F1 mjere kreće se između 0 i 1 gdje je 0 najlošiji rezultat, a 1 najbolji rezultat koji ukazuje da su sve vreće točno klasificirane. Najviša postignuta vrijednost ovog modela je 0.4766 i postignuta je u trećoj epohi što se poklapa sa grafom funkcije gubitka.

S obzirom na to da funkcija gubitka ne pada značajno od 4. epohe, možemo zaključiti da je model nakon treće epohe pretreniran i više griješi na skupu podataka za testiranje, zbog čega je F1 mjera manja.

Kao što je ranije spomenuto, na rezultate utječe velik broj lažno pozitivnih instanci slika, odnosno slika na kojima nije vidljivo odbacivanje, ali pripadaju vreći kojoj je dodeljena visoka razina odbacivanja radi drugih slika.



Slika 4.2. Mjera F1 nakon svake epohe

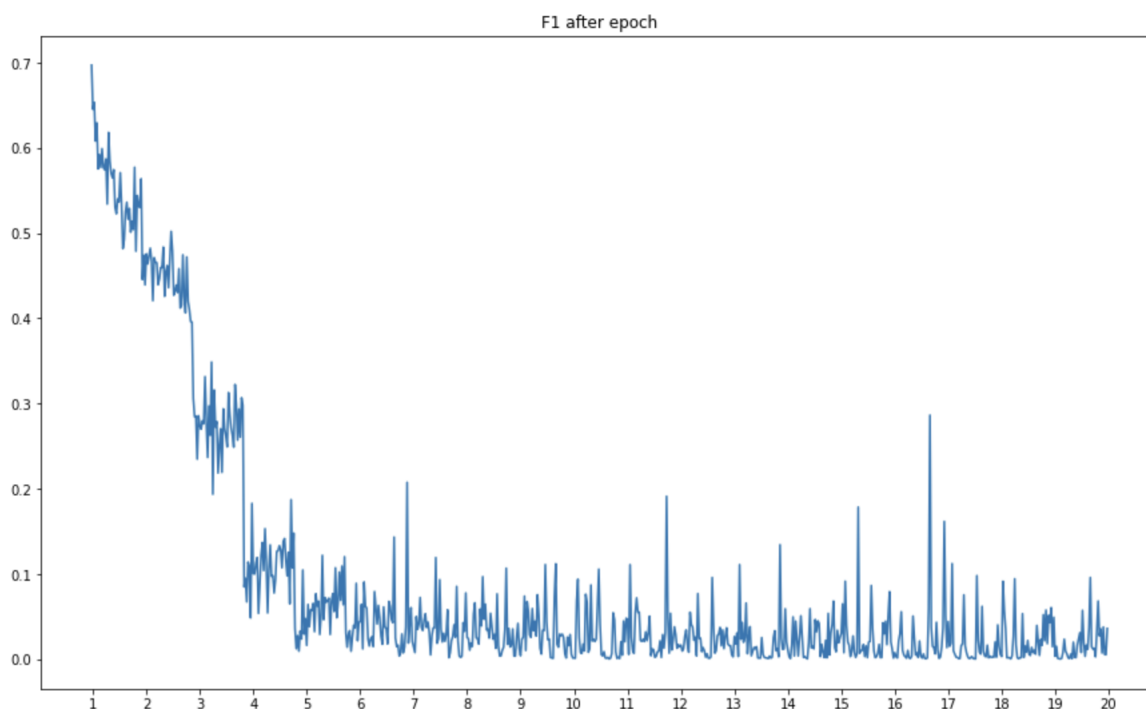
4.2. Neuronska mreža sa 2 klase

Isti model kao iz 4.1. treniran je na skupu podataka u kojem su klase grupirane na dvije klase. 0R i 1R grupirane su u grupu s nižom razinom odbacivanja, a 2R i 3R su grupirane u grupu s višom razinom odbacivanja. Broj slika u ovako razdvojenim grupama prikazan je u tablici 4.2.

Oznaka	Broj instanci	Postotak instanci
0R - 1R	15506	69.46%
2R - 3R	6816	30.53%

Slika 4.3. prikazuje vrijednost funkcije gubitka na modelu treniranom na skupu podataka sa dvije klase. Model je također treniran 20 epoha kao i prethodni.

Graf funkcije gubitka sličan je prethodnom s razlikom da inicijalna vrijednost funkcije gubitke kreće od 0.6972, što bi se dalo objasniti manjim brojem mogućih klasa. Prethodno je model mogao klasificirati sliku kao 3R iako je njena prava oznaka 2R što bi značilo pogrešnu klasifikaciju, no u slučaju s grupiranim klasama, ista slika može biti



Slika 4.3. Funkcija gubitka svakih 100 iteracija

ispravno klasificirana kao 2R - 3R.

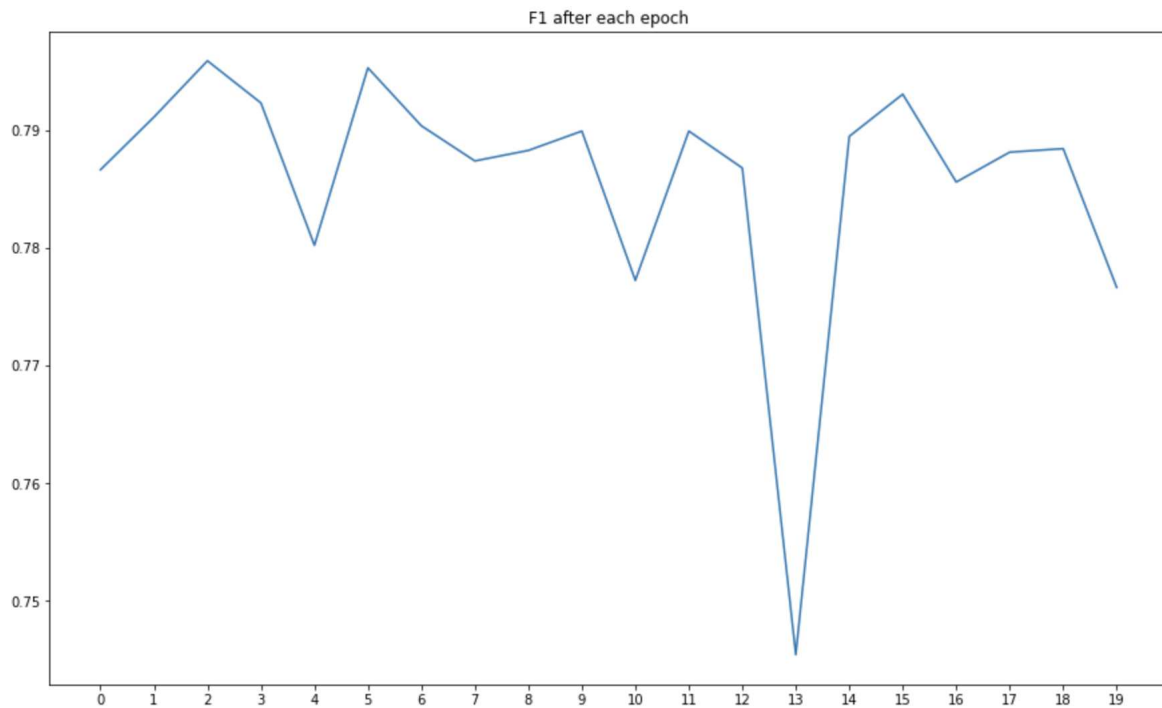
Kao i na prethodnom grafu, vidljivo je da nakon pete epohe, model ne napreduje i postaje pretreniran na postojećim slikama. Za razliku od prethodnog modela, u kasnijim epohama su vidljivi nagli skokovi vrijednosti funkcije gubitka do 0.3.

Mjera F1 najviša je nakon druge epohe, nakon čega pada, ali ne kao na modelu treniranom na skupu podataka s četiri klase. Također je bitno napomenuti kako je najveća vrijednost F1 mjere u ovom slučaju 0.7958 što je znatno više nego na prošlom modelu. To proizlazi iz toga što model može klasificirati vreću u manji broj klasa.

4.3. Multiple instance learning model

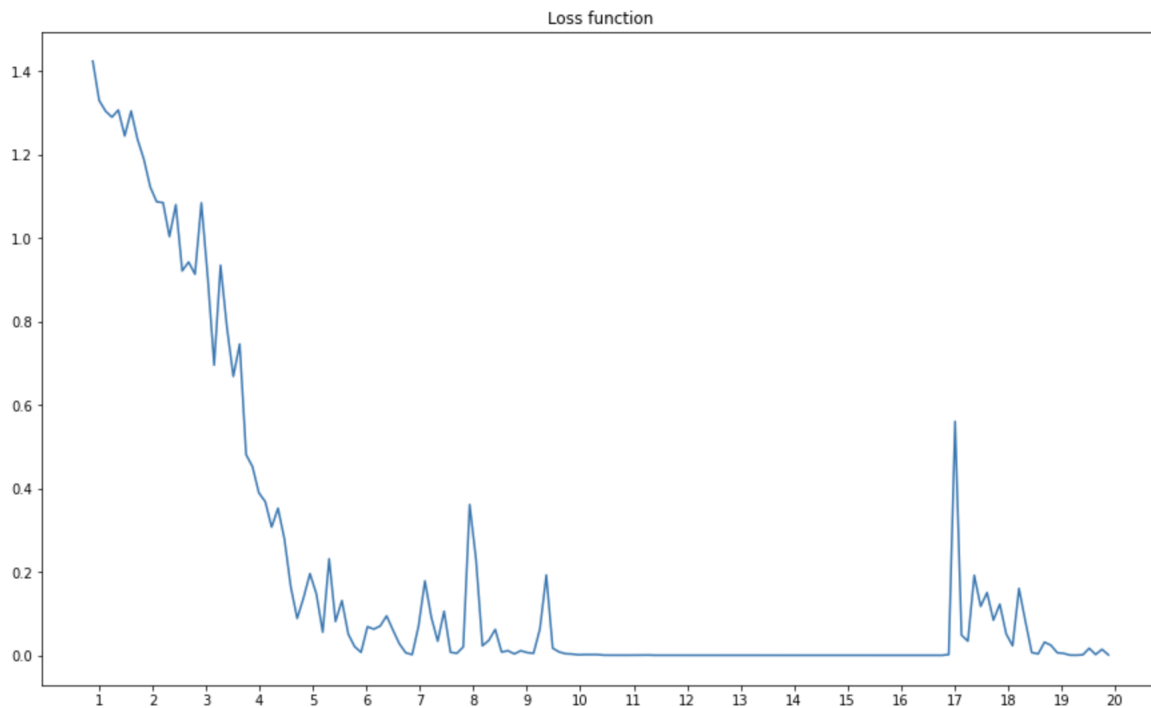
Zadnji trenirani model je neuronska mreža proširena s multiple instance learning metodom. Model je također treniran na dvadeset epoha i evaluiran na skupu za treniranje nakon svake.

Na slici 4.5. prikazana je prosječna vrijednost funkcije gubitka na svakih 100 vreća. Za



Slika 4.4. F1 mjera nakon svake epohe

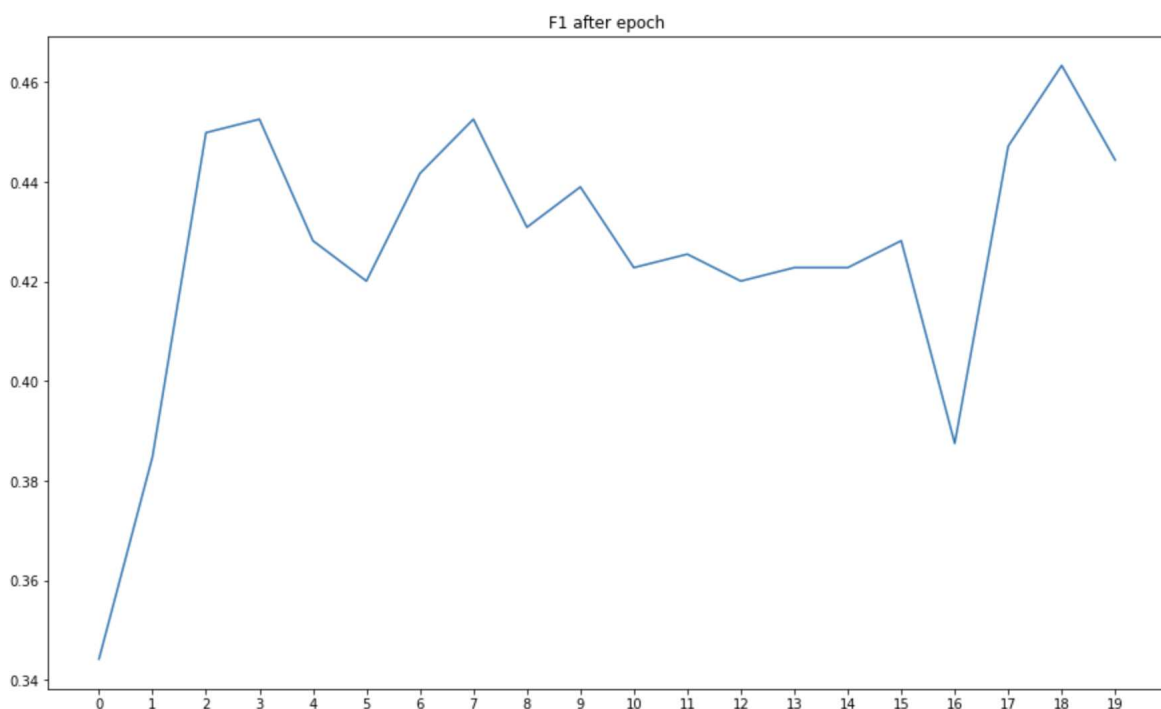
razliku od grafa funkcije gubitka prethodnog modela, graf na slici ima manje podataka s obzirom na to da vreća ima manje nego samih slika na kojima je treniran prošli model.



Slika 4.5. Funkcija gubitka svakih 100 iteracija

Vrijednost funkcije gubitka inicijalno kreće od 1.4243 i također se spušta do pete epohe kao i prošli modeli. Za razliku od vrijednosti funkcije gubitka kod prošlih modela, nakon desete epohe je vrijednost gubitka stabilno mala. Vrijednost je kod prethodnih modela bila između 0 i 0.1.

Od desete do sedamnaeste epohe je vrijednost stope učenje manja od nula, no u sedamnaestoj epohi je vrijednost skočila na 0.4. Nakon skoka, vrijednost ponovno ide prema nuli i do dvadesete epohe dođe do stabilne i niske vrijednosti. Ovakav skok je vjerojatno posljedica pretreniranosti modela. Ovakvi skokovi nisu prisutni kod prethodnih modela, no vrijednost njihove funkcije gubitka nije stabilna kao ona multiple instance learning modela.



Slika 4.6. F1 mjera nakon svake epohe

Slika 4.6. prikazuje F1 mjeru na modelu nakon svake epohe. F1 mjera je na kraju prve epohe 0.3442, te model dostiže 0.4526 nakon treće epohe. U sedamnaestoj epohi modela vidljiv je pad F1 mjere na 0.3875 što korelira s porastom vrijednosti funkcije gubitka vidljive na 4.5.

Na temelju rezultata vidljivo je da je model treniran na vrećama umjesto na poje-

dinačnim instancama stabilniji u pogledu vrijednosti funkcije gubitka u kasnijim epohama. S druge strane, kada se usporedi obična neuronska mreža i model proširen multiple instance learning metodom, MIL model nije ostvario značajno bolje rezultate. Moguća poboljšanja na modelu navedena su u poglavlju 5.

5. Moguća poboljšanja

Kako bi poboljšali performanse modela, moguće je dalje eksperimentirati s modelom što uključuje njegovu arhitekturu, hiperparametre i algoritme, te eksperimentirati sa skupom podataka za učenje.

5.1. Prilagodba modela

Spomenuta je arhitektura multiple instance learning modela i njegovi slojevi. Arhitekturu smo mogli mijenjati tako da dodajemo, mijenjamo ili mičemo slojeve mreže kako bi dobili model s najboljim performansama na skupu za treniranje.

S obzirom na to da se opisani MIL model sastoji od dva dijela, osim eksperimentiranja dijelom koji je neuronska mreža, moguće je eksperimentirati i drugim dijelom koji služi agregaciji oznaka vreće. Inicijalno je odabran Attention pooling, ali dostupne su i druge metode kao što su max pooling ili mean pooling. Problem ova dva operatora je što su unaprijed definirani i nije ih moguće trenirati za zasebne probleme. Uz eksperimentiranje slojevima koji čine prvi dio modela, također je moguće prilagođavati slojeve attention pooling dijela i time poboljšati performanse.

Trenirani model ima definirane hiperparametre i algoritme koje koristi pri treniranju. Pri treniranju MIL modela korišten je AdamW koji je ranije u radu spomenut. Moguće poboljšanje bilo bi biranje novog algoritma za ažuriranje parametara, no to često ne donosi velike razlike u performansama modela.

S druge strane AdamW algoritam ažurira parametre mreže na temelju svojih hiperparametara. Hiperparametri algoritma AdamW su stopa učenja, β_1 i β_2 . Za potrebe treniranja modela korištene su vrijednosti 0.01 za stopu učenja, a za β_1 i β_2 su ostavljene zadane vrijednosti iz paketa torch biblioteke 0.9, 0.999.

5.2. Prilagodba skupa podataka

U dijelu rada koji opisuje skup podataka 3.1., prikazana je tablica na koja prikazuje koliko instanci u skupu za učenje pripada kojoj klasi, i koliki udio u skupu podataka imaju klase.

Klase s manjom razinom odbacivanja imaju veći udio u skupu podataka što može utjecati na performanse modela. Za rješavanje tog problema, dio skupa za učenje mogao bi se generirati iz postojećih slika kako bi udio u skupu za treniranje bio sličan. Nove slike moguće je dobiti ponavljanjem postojećih, te dodavanjem šumova na postojeće slike.

6. Zaključak

Kroz rad je istražena tema dijagnosticiranja odbacivanja srca nakon transplantacije i kako je s pomoću strojnog učenja moguće detektirati odbacivanje na histološkim analizama.

Detaljno su objašnjene osnove histološke analize, uključujući metode prikupljanja i obrade uzoraka srčanog tkiva. Prikazano je kako histološke analize tkiva srca mogu otkriti znakove odbacivanja, te kako se ove analize interpretiraju kako bi se prepoznalo i klasificiralo odbacivanje. Na primjeru jedne histološke analize, pokazano je kako izgledaju stanice srca iz različitih kutova, te kako se odbacivanje manifestira kroz infiltrate, odnosno nakupine bijelih krvnih stanica u transplantatu.

S tehničke strane, rad opisuje neuronske mreže i multiple instance learning metodu. U sklopu rada implementirana je neuronska mreža specijalizirana za problem detekcije odbacivanja srca, koja je trenirana na dostupnom skupu podataka. Objašnjen je proces razvoja modela, uključujući prikupljanje i pripremu podataka, arhitekturu mreže, te postupke treniranja i validacije modela. Također, neuronska mreža je trenirana na skupu podataka čije su oznake grupirane u više i niže razine odbacivanja kako bi se s manje izlaznih klasa dobili bolji rezultati.

Neuronska mreža je zatim proširena metodom multiple instance learninga, koja omogućava efikasniju analizu kompleksnih podataka gdje su oznake dostupne samo na razini skupova instanci, a ne pojedinačnih instanci.

Napravljena je usporedba performansi neuronske mreže i modela proširenog multiple instance learning metodom. Prikazani su rezultati treniranih modela kao vrijednosti funkcije gubitka i mjere F1. Rezultati modela međusobno su validirani na podacima na kojima nisu trenirani, te su rezultati validacija međusobno uspoređeni.

Kraj rada opisuje moguća poboljšanja na modelima koja bi se mogla naknadno implementirati kao što je mijenjanje arhitekture i hiperparametara modela, ali i prilagođavanje skupa podataka.

Zaključno, rad istražuje primjenu strojnog učenja i računalnog vida pri dijagnostici-ranju odbacivanja srca nakon transplantacije.

Literatura

- [1] I. Aleksander i H. Morton, *An Introduction to Neural Computing*. Chapman and Hall, 1990.
- [2] R. Gate, “Resarch gate”. [Mrežno]. Adresa: https://www.researchgate.net/figure/Artificial-neural-network-architecture-ANN-i-h-1-h-2-h-n-o_fig1_321259051
- [3] M.-A. Carbonneau, V. Cheplygina, E. Granger, i G. Gagnon, “Multiple instance learning: A survey of problem characteristics and applications”, *arXiv preprint arXiv:1612.03365*, 2018. [Mrežno]. Adresa: <https://arxiv.org/pdf/1612.03365>
- [4] P. Maia, “Nilg ai”. [Mrežno]. Adresa: <https://nilg.ai/202105/an-introduction-to-multiple-instance-learning/>
- [5] Wikipedia contributors, “Bioptome — Wikipedia, The Free Encyclopedia”, 2024., [Online; accessed 18-June-2024]. [Mrežno]. Adresa: <https://en.wikipedia.org/wiki/Bioptome>
- [6] PyTorch contributors, “torch.optim.AdamW”, 2024., [Online; accessed 18-June-2024]. [Mrežno]. Adresa: <https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html>
- [7] M. Ilse, J. M. Tomczak, i M. Welling, “Attention-based deep multiple instance learning”, *arXiv preprint arXiv:1802.04712*, 2018. [Mrežno]. Adresa: <https://arxiv.org/pdf/1802.04712>
- [8] P. Liu, L. Ji, X. Zhang, i F. Ye, “Pseudo-bag mixup augmentation for multiple instance learning-based whole slide image classification”, *IEEE Transactions on*

Medical Imaging, sv. 43, br. 5, str. 1841–1852, 2024. <https://doi.org/10.1109/TMI.2024.3351213>

- [9] B. Dalbelo Bašić i J. Šnajder, “Vrednovanje klasifikatora”, 2021., [Online; accessed 18-June-2024]. [Mrežno]. Adresa: https://www.fer.unizg.hr/_download/repository/SU-12-VrednovanjeKlasifikatora.pdf

Sažetak

Predtreniranje dubokih neuronskih mreža na analizu medicinskih slika

Petar Cvitanović

Nakon transplantacije srca, moguća su odbacivanja transplantanta. Proces odbacivanja presađenog organa bitno je detektirati i tretirati ako je potrebno. Za tu svrhu, biopsijom i histološkom analizom prikupljene su slike srca nakon transplantacije na kojima je vidljiva struktura srčanog tkiva. Kroz rad je implementiran model neuronske mreže treniran na skupu slika srca kako bi mogao prepoznati odbacivanje i klasificirati ga u jednu od četiri klase. Model neuronske mreže također je proširen multiple instance learning metodom koja dozvoljava da model treniramo na skupu slika, umjesto pojedinačnim slikama, za što imamo pripremljen skup podataka. Modeli s i bez multiple instance learning metode su nakon treniranja evaluirani i uspoređeni po svojim F1 mjerama i vrijednostima funkcije gubitka.

Ključne riječi: strojno učenje, računalni vid, neuronska mreža, multiple instance learning, transplantacija srca

Abstract

Pretraining deep neural networks for medical image analysis

Petar Cvitanović

After a heart transplant, graft rejections are possible. The process of rejection of the transplanted organ is essential to detect and treat if necessary. For this purpose, biopsy and histological analysis were used to collect images of the heart after transplantation, showing the structure of the heart tissue. The paper explains the implementation of a neural network model trained to recognize rejection and classify it into one of four classes depending on the rejection severity. The neural network model is also extended with the multiple instance learning method, which allows us to train the model on a set of images instead of individual images, for which we have a prepared data set. Models with and without the multiple instance learning method were evaluated after training and compared according to their F1 measures and loss function values.

Keywords: machine learning, computer vision, neural network, multiple instance learning, heart transplant