

# Razvoj metode temeljene na strojnom učenju za pronalaženje sličnih prijeloma zapešća iz pedijatrijskih medicinskih rendgenskih slika

---

Frlan, Anton

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:648000>

Rights / Prava: [Attribution 4.0 International](#) / [Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2024-08-22**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI  
TEHNIČKI FAKULTET  
Diplomski sveučilišni studij računarstva

Diplomski rad

Razvoj metode temeljene na strojnom  
učenju za pronalaženje sličnih prijeloma  
zapešća iz pedijatrijskih medicinskih  
rendgenskih slika

Rijeka, studeni 2022.

Anton Frlan  
0069083010

SVEUČILIŠTE U RIJECI  
**TEHNIČKI FAKULTET**  
Diplomski sveučilišni studij računarstva

Diplomski rad

**Razvoj metode temeljene na strojnom  
učenju za pronalaženje sličnih prijeloma  
zapešća iz pedijatrijskih medicinskih  
rendgenskih slika**

Mentor: prof.dr.sc. Ivan Štajduhar

Rijeka, studeni 2022.

Anton Frlan  
0069083010

Umjesto ove stranice umetnuti zadatak  
za završni ili diplomski rad

## Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad pod vodstvom mentora prof. dr. sc. Ivana Štajduhara.

Rijeka, studeni 2022.

-----  
Anton Frlan

# Zahvala

Zahvaljujem svojoj obitelji i prijateljima na podršci, društvu i svoj zabavi koju smo podijelili tijekom studija.

Zahvaljujem svojem mentoru prof. dr. sc. Ivanu Štajduharu na strpljenju, razumijevanju, svim savjetima i na svom prenesenom znanju.

Posebno zahvaljujem asist. Franku Hržiću na svim idejama i pomoći prilikom izrade ovog rada i istraživanju metoda strojnog učenja.

Za kraj zahvaljujem svim profesorima i asistentima na svim predavanjima i vježbama tijekom kojih su pokazali svoje beskonačno strpljenje i razumijevanje prema studentima.

# Sadržaj

<b>Pojmovnik</b>	<b>i</b>
<b>1 Uvod</b>	<b>1</b>
1.1 Pregled literature . . . . .	3
<b>2 Definicija problema i metodologija</b>	<b>4</b>
2.1 Skup podataka . . . . .	4
2.1.1 Svrstavanje fraktura . . . . .	6
2.2 Definicija problema detekcije sličnih fraktura . . . . .	9
2.3 Razvijeni koncept za detekciju sličnih fraktura . . . . .	10
2.3.1 Ekstraktor značajki . . . . .	10
2.3.2 Podudaranje značajki . . . . .	16
<b>3 Rezultati</b>	<b>23</b>
3.1 Metrike . . . . .	23
3.1.1 Ekstraktor značajki, FE . . . . .	23
3.1.2 Podudaranje značajki . . . . .	24
3.2 Rezultati . . . . .	26
3.2.1 Ekstraktor značajki . . . . .	26
3.2.2 Podudaranje značajki . . . . .	28

## *Sadržaj*

<b>4</b>	<b>Diskusija</b>	<b>44</b>
<b>5</b>	<b>Zaključak</b>	<b>46</b>
5.1	Budući rad . . . . .	47
	<b>Bibliografija</b>	<b>48</b>
	<b>Pojmovnik</b>	<b>54</b>
	<b>Sažetak</b>	<b>55</b>
	<b>Popis slika</b>	<b>57</b>
	<b>Popis tablica</b>	<b>59</b>
<b>A</b>	<b>Enkoder dekodeer arhitekture</b>	<b>61</b>
<b>B</b>	<b>Dodatni primjeri podudaranja slika</b>	<b>64</b>



# Poglavlje 1

## Uvod

Digitalizacijom i razvojem baza podataka koje sadrže korisne informacije javlja se potreba za razvojem metoda pronalaženja relevantnih informacije prema danom kontekstu. U medicini, jedan od oblika kako informacija jest i slika. U svrhu pronalaženja samo najrelevantnijih slika kvalitetna metoda pronalaženja sličnih slika (*engl. Image Retrieval, IR*) je od neophodne važnosti. Glavna ideja IR modela je pronalaženje i sortiranje slika na bazi sličnosti promatrane slike (slika na temelju koje se traže sličnosti) s drugim slikama.

Postoje dvije glavne podjele IR modela: pretraga podataka za slikama sličnih anotacija (*engl. Text Based Image Retrieval, TBIR*) i pomoću značajki slika (*engl. Content Based Image Retrieval, CBIR*) [1, 2, 3, 4, 5]. Pronalaženje sličnih slika pomoću anotacija zahtjeva anotiranje slika što može biti resursno iscrpno bilo da se radi o ljudskim resursima ili o materijalnim. Ponekad, s velikim bazama podataka koje zahtijevaju stručno znanje, može biti i vremenski zahtjevan posao. S druge strane, CBIR temelji se isključivo na ekstrakciji korisnih značajki slike. Generalni okvir CBIR modela sastoji se od:

- Sheme prosljeđivanja slika (*engl. Feedforward Scheme, FS*)
- Ekstraktora značajki (*engl. Feature Extractor, FE*)
- Podudaranja značajki (*engl. Feature Matching, FM*)

CBIR pretraga može biti temeljena na bilo kojim značajkama koje se mogu izlučiti

## Poglavlje 1. Uvod

iz slike.

Neki od učestalijih jednostavnih ekstraktora značajki su bazirani na boji [6, 7, 8] i frekvencijskim transformacijama [7, 9, 10, 11]. Međutim, popularni su i napredniji ekstraktori značajki poput transformacija značajki nepromjenjivih mjerila (*engl. Scale-Invariant Feature Transform, SIFT*) [12, 13, 14], torba vizualnih riječi (*engl. Bag of Visual Words, BoVW*) [15, 14] i konvolucijske neuronske mreže (*engl. Convolutional Neural Network, CNN*) [16, 12]. Nakon stvaranja prostora oznaka (*engl. Feature Space*) pomoću ekstraktora značajki, treba izračunati semantičku udaljenost slika. Semantička udaljenost slika može se izračunati korištenjem tehnika za računanje sličnosti poput Euklidske udaljenosti (*engl. Euclidean Distance*), Mahalanobisove udaljenosti (*engl. Mahalanobis Distance*), učenja metrika (*engl. Metric Learning*) [16] i drugih tehnika. CBIR se može podijeliti na pronalaženje slika na razini kategorije (*engl. Category Level Image Retrieval, CIR*) [9] i pronalaženje slika na razini zasebnog slučaja (*engl. Instance Level Image Retrieval, IIR*) [17]. IIR vraća sliku iste instance s promatrane slike, dok CIR vraća sliku iste kategorije, odnosno klase.

U ovom radu se predstavlja razvijena metoda strojnog učenja za pronalaženje sličnih fraktura zgloba s pedijatrijskih medicinskih rendgenskih slika. Koristeći rekonstrukcijski autoenkoder za ekstrakciju slika, sačuvati će se sve bitne informacije podatka te će se smanjiti njegova veličina. Za podudaranje značajki isprobano je nekoliko dubokih umjetnih neuronskih mreža (*engl. Deep Artificial Neural Network, DNN*) arhitektura. Podatci se sastoje od 20 330 lateralnih i anteroposteriornih projekcija ruku s frakturama te sadrži 43 klase. Ostatak rada je organiziran u sljedećim cjelinama:

1. Pregled literature - diskusija srodne literature u medicini
2. Metodologija i definicija problema - opis problema, korištenih podataka, FE i FM
3. Rezultati - prezentacija korištenih metrika i rezultata
4. Diskusija - osvrt na problematiku, treniranje i rezultate
5. Zaključak - pregled rada, zaključak i buduće nadogradnje

## 1.1 Pregled literature

Pregledom dostupne literature uočeno je da CBIR postiže dobre rezultate i na medicinskim slikama. Sustavna i temeljita istraživanja primjene te metode u medicini u posljednjih dvadeset godina [18, 19, 20, 21] dovela su do značajnog napretka, te je razvijena nova verzija CBIR metode, pronalaženje sličnih medicinskih slika pomoću značajki (*engl. Content Based Medical Image Retrieval, CBMIR*). Veliki broj novo-generiranih podataka rezultira i potrebom za sustavima arhiviranja slika i komunikacija (*engl. Picture Archiving and Communication systems, PACS*) [20, 21]. Na odjelima poput radiologije i kardiologije gdje se nove slike stvaraju svakodnevno, CBMIR ima veliki potencijal kao pomoćni alat tijekom donošenja odluka omogućujući lagan i brz pregled sličnih bivših slučajeva [19, 20].

Međutim, semantička udaljenost (*engl. Semantic Gap*) među slikama narušava performanse modela ako su korištene samo značajke niskih razina (na primjer rubovi), koje ne sadrže bitne informacije o promatranom fenomenu (bolesti). U radu [22] koristi se duboka konvolucijska umjetna neuronska mreža (*engl. Deep Convolutional Neural Network, DCNN*) za ekstrakciju značajki umanjujući problem semantičke udaljenosti. Isti pristup se može vidjeti u radu [23] koji koristi DCNN model CNN autoenkodera kao ekstraktor značajki. Sličan pristup se može vidjeti u radu [24] u kojem je CNN autoenkoder korišten za uklanjanje šuma iz podataka.

Radovi [25, 22] pokazuju da je klasifikacija validan pristup učenja modela u području CBMIR. Kako bi se izračunala sličnost među slikama treba se iskoristiti određena matematička funkcija za računanje udaljenosti (npr. Euklidska udaljenost) [22]. U trenutku pisanja ovog rada ne postoje uspješne CBMIR metode isprobane na frakturama zgloba s pedijatrijskih medicinskih rendgenskih slika te će ovo biti prvi rade te vrste. Korištena baza podataka je prikupljena u bolnici u Grazu te objavljena nakon anonimizacije svih podataka [26].

# Poglavlje 2

## Definicija problema i metodologija

U ovom poglavlju prvo će se u poglavlju *Skup podataka* prikazati podatci i njihova raspodjela zajedno s opisom sadržaja i provedenom predobradom podataka. Zato, će se u poglavlju *Definicija problema detekcije sličnih fraktura* predstaviti problematika pronalaženja sličnih fraktura zgloba s pedijatrijskih medicinskih rendgenskih slika i isprobana rješenja temeljena na strojnom učenju. Na kraju će se u poglavlju *Razvijeni koncept za detekciju sličnih fraktura* predstaviti glavne dijelove metode. U svakom dijelu metode isprobano je nekoliko različitih algoritama čije će korištenje biti obrazloženo.

### 2.1 Skup podataka

Korišteni podatci su pedijatrijske medicinske rendgenske slike fraktura zgloba ruke. Sve slike su monokromatske. Baza podataka je u trenutku preuzimanja s ovlaštene stranice sadržala 20 330 rendgenskih slika fraktura zgloba ruke [26]. Svaka slika ima nekoliko svojstava koja ju opisuju. Svojstva koja su razmatrana u sklopu ovog rada su: projekcija (*engl. Projection*), strana (*engl. Side*) i klasa ili tip frakture.

Projekcija predstavlja topološku os s koje je ruka snimana rendgenskim zrakama. Moguće pozicije su:  $\text{projection}_{ap}$  (slika 2.1a 2.1b),  $\text{projection}_{lat}$  (slika 2.1c 2.1d) i  $\text{projection}_{oblique}$ .  $\text{Projection}_{ap}$  znači da je ruka tijekom rendgenskog snimanja bila u položaju s dlanom prema stolu te nadlanicom prema rendgenskoj cijevi. Svojstvo

## Poglavlje 2. Definicija problema i metodologija

$projection_{lat}$  je pozicija u kojoj je bočna strana ruke položena na stol dok je druga bočna strana ruke usmjerena prema rendgenskoj cijevi. U svrhe ovog rada pozicija  $projection_{oblique}$  nije korištena radi malog broja primjeraka te su sve slike s navedenom pozicijom izbačene iz baze podataka. Na slici 2.2 prikazana je raspodjela rendgenskih slika fraktura zgloba ruke po projekcijama 2.2b i po strani 2.2a.

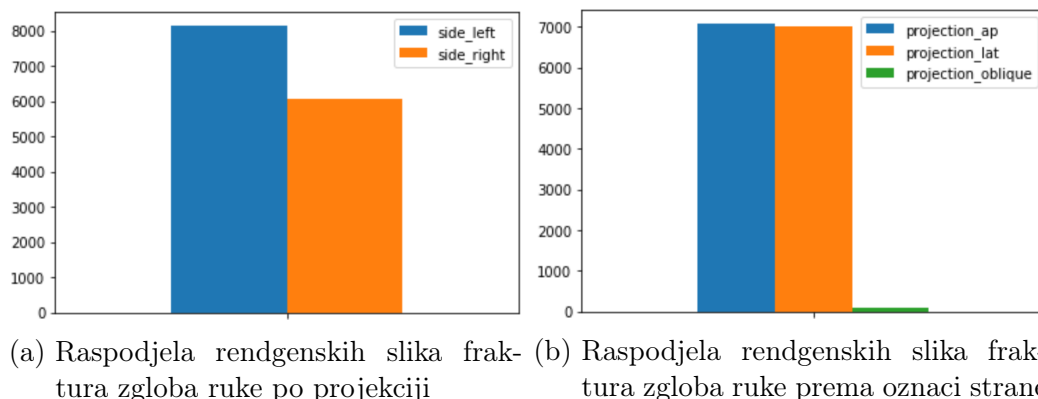
Svojtvo strana je pokazatelj ruke koja je snimana. Može sadržati vrijednosti  $side_{left}$  i  $side_{right}$ . Vrijednost  $side_{left}$  znači da je snimljena lijeva ruka na slici kao što je prikazano na slikama 2.1a i 2.1c. S njihovih desnih strana nalaze se primjeri vrijednost  $side_{right}$  odnosno slike desnih ruku (slike 2.1b i 2.1d).



Slika 2.1 Primjeri slika s različitim vrijednostima projekcije i strane.

Klase fraktura su tipovi fraktura koji su prisutni na snimljenoj ruci. Originalna baza podataka u trenutku skidanja sa stranice je sadržavala 43 klase fraktura i 3 nat-

## Poglavlje 2. Definicija problema i metodologija



Slika 2.2 Raspodjela rendgenskih slika fraktura zgloba ruke po projekciji i oznaci strane.

klase fraktura. Pripadnost natklasi frakture se može iščitati iz velikog tiskanog slova u klasi frakture. Nekolicina klasa je bila tvorevina pogreške prilikom utipkavanja. Na primjer "23u-E/7<" umjesto "23u-E/7" i "23r-M2.1" umjesto "23-M/2.1". Jedine iznimke tome su bile klase koje se odnose na tip fraktura "8620bbff3ebaae183964c", "2f99822f0c6ae7ebf" i "69a677a94dadce5d3c416b39210b82" za koje se nije uspjelo pronaći objašnjenje kako su nastale. Sve slike s krivo klasificiranim frakturama su bile izbačene iz baze podataka. Osim krivo klasificiranih fraktura izbačene su i sve klase za koje postoji manje od 100 primjeraka. Nakon navedenog pročišćivanja baze podataka preostale podatci su se sastojali od devet klasa i tri natklase. Devet klasa i njihovu količinu može se vidjeti u tablici 2.1, a tri natklase i njihovu količinu u tablici 2.2.

### 2.1.1 Svrstavanje fraktura

Pošto je predmet istraživanja ovog rada pronalaženje sličnih fraktura, iz slika se mora izlučiti sve označene frakture i odrediti im klase. Svaka rendgenska slika ima jednu do četiri frakture. Frakture mogu biti bilo koje klase te više fraktura može biti iste klase. Kako bi se odredila klasa svake frakture razvijen je jednostavni sustav temeljen na pravilima (*engl. Rule Based System*). Njime se svakoj označenoj frakturi pridodjeljuje klasa iz anotacija slike kojoj ta fraktura pripada.

Poglavlje 2. Definicija problema i metodologija

Naziv klase	Broj primjeraka
23r-M/3.1	5172
23r-M/2.1	3627
23u-E/7	3209
23u-M/2.1	2330
23-M/3.1	1667
23r-E/2.1	1430
23-M/2.1	1410
22r-D/2.1	162
23r-E/2.1	106

Tablica 2.1 Prikaz preostalih klasa fraktura u bazi podataka nakon pročišćavanja.

Naziv natklase	Broj primjeraka
E	4745
M	14206
D	162

Tablica 2.2 Prikaz preostalih natklasa fraktura u bazi podataka nakon pročišćavanja.

U slučaju da slika sadrži samo jednu klasu, sve frakture označene na toj slici će biti navedene klase. Ako rendgenska slika sadrži više od jedne klase fraktura, mora se razaznati koja fraktura pripada kojoj klasi. Prvo se provjeri je li slikana lijeva ili desna ruka. Pošto su lijeva i desna ruka simetrične, logika je napravljena za desnu ruku te je rezultat zrcaljen za lijevu ruku.

Nakon provjere radi li se o lijevoj ili desnoj ruci, klase fraktura se dijele na lijevu odnosno desnu frakturu. Klase fraktura može se podijeliti u tri skupine: fraktura ulne, fraktura radiusa i fraktura radiusa i ulne (radi lakšeg raspoznavanja ovu klasu se naziva frakturom obje kosti). Radius i ulna su lijeva i desna kost desne ruke (slika 2.3b), odnosno desna i lijeva kost lijeve ruke (slika 2.3a). Prilikom usporedbe, osim razlike u poziciji primijetit će se i razlika u debljini, to jest radius je deblja kost od ulne. Ako se isti tip frakture nalazi i na radiusu i na ulni to se smatra frakturom obje kosti. Stoga se ulnu svrstava pod frakturu desne kosti, radius pod frakturu lijeve kosti a frakturu obje kosti se svrstava i pod frakturu lijeve kosti i pod frakturu desne kosti. Time se dobiva dva niza s klasama fraktura koji govore na kojoj polovici

## Poglavlje 2. Definicija problema i metodologija



(a) Ulna i radius na lijevoj ruci



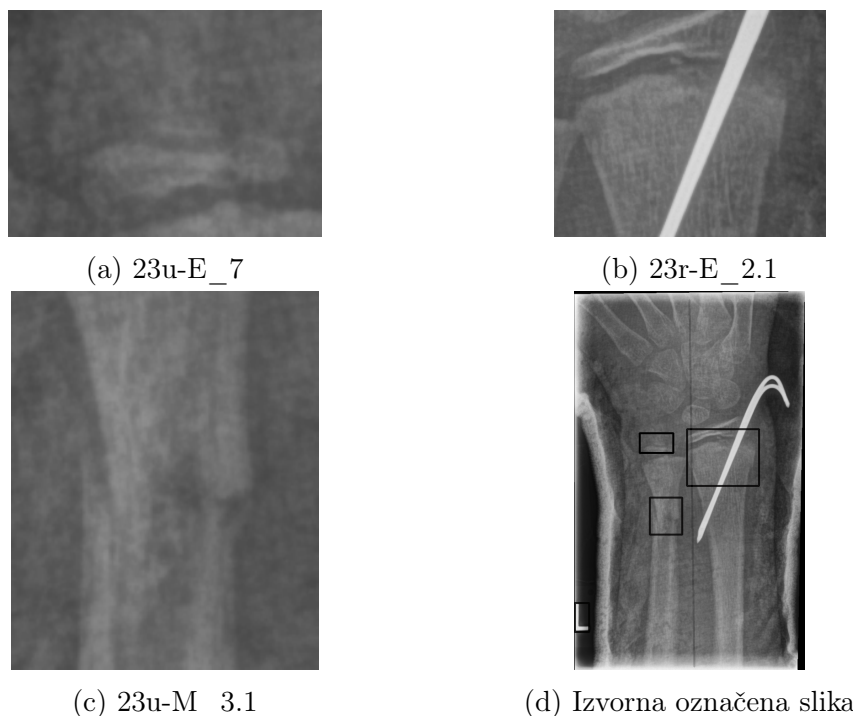
(b) Radius i ulna na desnoj ruci

Slika 2.3 Prikaz kostiju radius i ulna na lijevoj i desnoj ruci.

ruke se nalazi koja klasa frakture. Zatim se sortira frakture označene na rendgenskoj slici u lijevu i desnu frakturu. Za svaku frakturu na slici se provjeri kojoj je rub najudaljeniji od centra rendgenske slike. Potom se fraktura svrsta u odgovarajući niz ovisno o tome nalazi li se lijevo ili desno od centra rendgenske slike. Ako je ruka u  $projection_{lat}$  vidljiv je samo radius, te se sve frakture na slici svrstavaju u frakturu lijeve kosti.

Na kraju sortiranja označenih fraktura, provjerava se je li ruka na rendgenskoj slici lijeva te ako je, zamijeni se nizove lijeve i desne frakture. Zadnji korak u svrstavanju fraktura je sortiranje fraktura i klasa fraktura po visini u odnosu na ruku. Visinu frakture se dobije po visini centra frakture na slici, a visinu klase frakture se dobije iz njenog imena. Natklase 'E' su najviše, natklase 'M' su ispod njih, a natklase 'D' su najniže. Sada se svaka fraktura može svrstati u njenu klasu pomoću sortiranih nizova. Na slici 2.4 može se vidjeti primjer izlaza algoritma svrstavanja fraktura. Na slici 2.4d je prikazan rendgen lijeve ruke u  $projection_{lat}$ . Na slikama 2.4a, 2.4b i 2.4c su prikazane točno spojene klase fraktura i frakture.





Slika 2.4 Primjeri razvrstavanja klasa fraktura po označenim frakturama.

## 2.2 Definicija problema detekcije sličnih fraktura

Problematika nalaženja sličnih fraktura leži u potrebi za razlikovanjem sličnih fraktura različitih klasa te stvaranjem što veće međuklasne udaljenosti (*engl. Inter Class Distance*) i što manje unutar klasne udaljenosti (*engl. Intra Class Distance*). Unutar klasna udaljenost je udaljenost među podacima iste klase, dok je međuklasna udaljenost udaljenost podataka različitih klasa. Jedan od čestih problema na koji se može naići je manjak anotiranih podataka ili nepotpune anotacije. U nekim slučajevima, iako postoje anotacije ruke s prisutnim frakturama, nije određeno koja fraktura spada pod koju klasu.

Cilj ovog istraživanja je uspješna implementacija algoritma za povratak slika na razini klase. Osim u CIR svrhe, implementirani algoritam bi se mogao koristiti kao objašnjivi klasifikator (*engl. Explainable Classifier*). Koristeći učestalost klasa među najbližim frakturama kao vjerojatnosti, može se na objašnjiv način odrediti najvjerojatnija klasa. Primjera radi neka se odluka bazira na deset najbližijih slika.

## Poglavlje 2. Definicija problema i metodologija

Kada se napravi nova slika, provjeri se i učestalost klasa. Ako se u deset klasa nalazi šest klasa M i četiri klase D algoritam će tvrditi da je nova slika klase M sa sigurnošću od 60% te klasa D sa sigurnošću od 40%. Navedeni algoritam bi mogao biti korišten kao pomoćni alat doktora na raznim zavodima (traumatologiji, radiologiji, kardiologiji) pri klasifikaciji fraktura pokazujući im prijašnje najslabije frakture.

## 2.3 Razvijeni koncept za detekciju sličnih fraktura

Metoda detekcije sličnih fraktura sastoji se od dvije komponente:

1. Ekstraktor značajki (*engl. Feature Extractor, FE*)
  - Enkoder
  - Dekoder
  - Korišteni hiperparametri
2. Podudaranja značajki (*engl. Feature Matching, FM*)
  - Model sličnosti
  - Korišteni hiperparametri

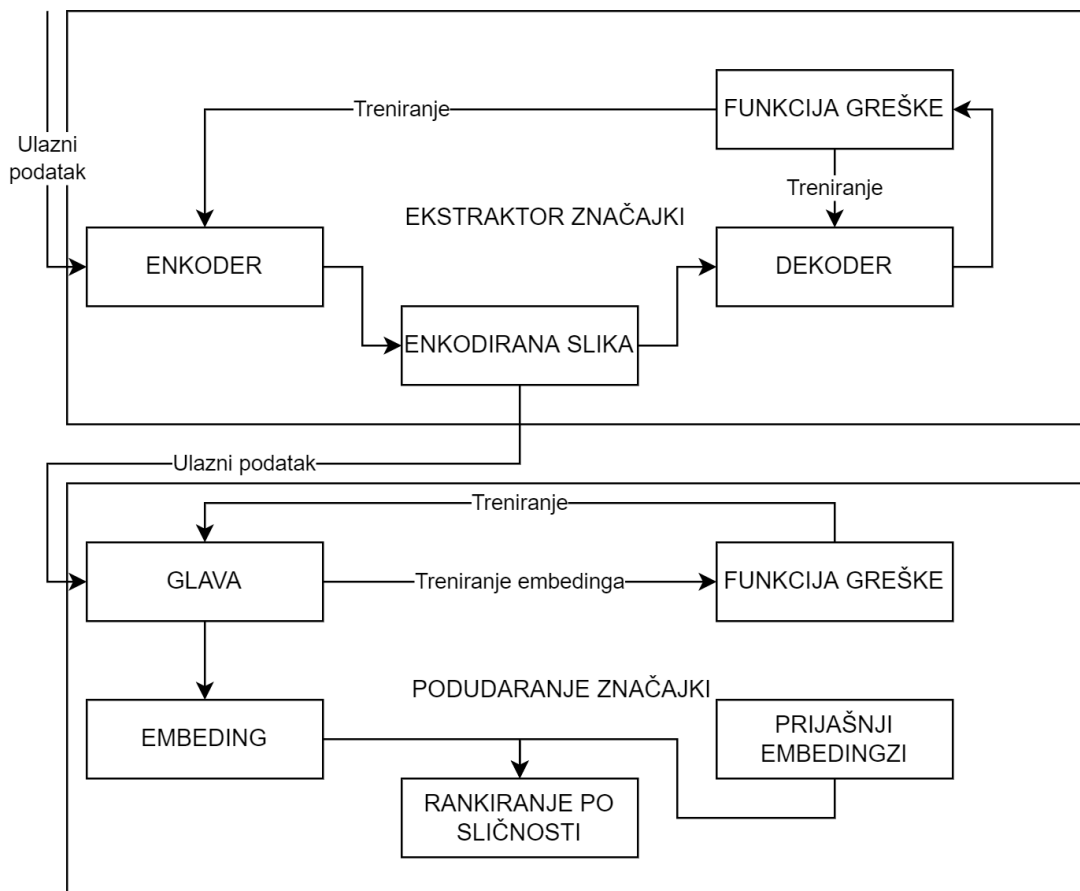
Iz slike se prvo ekstrahiraju značajke pomoću ekstraktora značajki kako bi se izvukle sve bitne informacije. Zatim su ekstrahirane značajke prosljeđene modelu podudaranja značajki koji izvršava podudaranje značajki i rangiranje slika po sličnosti. Navedena struktura prikazana je na slici 2.5. Svaka komponenta i potkomponenta razvijenog koncepta bit će objašnjene u nadolazećim potpoglavljima.

### 2.3.1 Ekstraktor značajki

Ekstraktor značajki je metoda reduciranja/transformiranja značajki tako da reducirane/transformirane značajke sadrže isključivo korisne informacije. Njen cilj je iz originalnog podatka izvući značajke koje su što informativnije i po mogućnosti ne ponavljajuće i međusobno ne korelirane. Koristi se iz mnogobrojnih razloga, od ko-

## Poglavlje 2. Definicija problema i metodologija

Slika frakture



Slika 2.5 Koncept za detekciju sličnih frakture.

jih su neki: smanjivanje dimenzionalnosti, brisanje šuma odnosno štetnih značajki, stvaranje raznolikijih i informativnijih značajki iz postojećih značajki [27, 28, 29].

U ovom radu, kao ekstraktor značajki, korištene su umjetne neuronske mreže (*engl. Artificial Neural Network*) sa svojstvima enkodera i dekodera [30]. Ideja njihovog korištenja je da se podatak kompresira što je više moguće, dok se u isto vrijeme zadrži što više informacija iz originalnog podatka. Količinu zadržanih informacija može se provjeriti korištenjem rekonstrukcije i uspoređivanjem razlike između rekonstruirane i originalne slike.

Podatci u ekstraktoru značajki su grupirani u dvije grupe: podatci za treniranje

## Poglavlje 2. Definicija problema i metodologija

(*engl. Training Set*) i podatci za validaciju (*engl. Validation Set*). Podatci za treniranje su podatci s kojima se trenira model strojnog učenja, dok se s validacijskim podacima provjerava validnost odabira hiperparametra modela. Podjela podataka između podataka za treniranje i validaciju je 80% i 20% odnosno 13 845 podataka za treniranje i 3464 podataka za validaciju.

### Enkoder i dekodeer

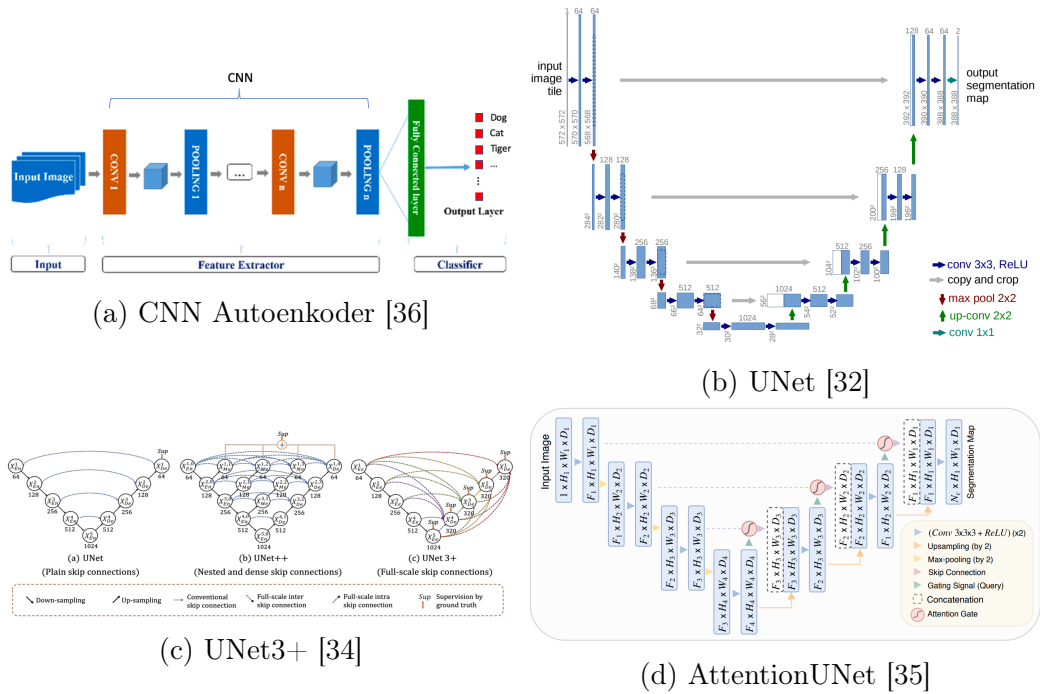
Enkoder i dekodeer je model strojnog učenja koji se sastoji od dvije umjetne neuronske mreže. Prva umjetna neuronska mreža, zvana enkoder, služi za transformaciju podatka u vektor manje dimenzionalnosti. Korištenjem druge umjetne neuronske mreže, zvane dekodeer, dobiveni vektor se može transformirati iz nižedimenzionalnog prostora u višedimenzionalni prostor. To je moguće jer su enkoder i dekodeer zapravo modeli identičnih arhitektura, ali primaju informacije u suprotnim smjerovima. Radi opisanog svojstva enkoderi i dekodeeri su vrlo često korišteni u područjima poput procesiranja prirodnog govora ili rekonstrukcija slike. Isprobani enkoderi i dekodeeri su:

- CNN Autoenkoder (*engl. CNN Autoencoder*) [31]
- UNet [32]
- UNet3+ [33, 34]
- AttentionUNet [35]

Arhitekture su prikazane na slikama: 2.6 i A.

CNN Autoenkoder je najjednostavnija isprobana enkoder dekodeer mreža. Sastoji se od CNN i AVG slojeva u enkoderu (slika A.1) te CNN slojeva i slojeva povećanja uzorkovanja (*engl. Upsampling*) u dekodeer. UNet koristi istu ideju kao CNN Autoenkoder i unaprjeđuje ju pomoću veze preskakivanja (*engl. Skip Connections*) [37]. veze preskakivanja uzima izlaz jednog sloja i kopira ga na ulaz drugog sloja. U UNet modelu izlaz svakog CNN bloka enkodera je kopiran na ulaz istog CNN bloka enkodera. Na primjer izlaz prvog CNN bloka enkodera je kopiran na ulaz zadnjeg CNN bloka dekodeera, izlaz drugog CNN bloka enkodera je kopiran na ulaz predzadnjeg CNN bloka dekodeera i tako dalje. Pogleda li se slika A.2 primijetit će se da arhitektura sličí velikom slovu "U" po čemu je mreža i dobila ime. UNet3+ nadograđuje

## Poglavlje 2. Definicija problema i metodologija



Slika 2.6 Korišteni enkoder dekodler modeli.

UNet arhitekturu uvodeći vezu preskakivanja sa svih razina enkodera na sve razine dekodera. Također se uvodi učenje pod nadzorom na svakoj razini dekodera sa sigmoid aktivacijskom funkcijom. AttentionUNet koristi drugačiji pristup nadogradnje UNet-a od UNet3+. AttentionUNet ne spaja slojeve pomoću veze preskakivanja, već u tu svrhu koristi mehanizam pažnje (*engl. Attention*) na svaku vezu preskakivanja.

### Korišteni hiperparametri

U strojnom učenju hiperparametar je svaki parametar čijom promjenom možemo utjecati na sposobnost učenja modela te njegove performanse. U ekstratoru značajki koristile su se dvije vrste hiperparametara: parametri koji mijenjaju podatak te parametri koji mijenjaju model.

Pod hiperparametre koji utječu na podatak smatra se: augmentacijske metode (*engl. Augmentation Methods*), oblik (*engl. Shape*) i metode miješanja podataka (*engl. Data Shuffle Method*).

## Poglavlje 2. Definicija problema i metodologija

Augmentacijske metode su tehnike mijenjanja podatka tijekom treniranja u svrhu bolje generalizacije modela. Korištene augmentacijske metode su: dodavanje nasumičnog šuma, rotiranje slike, zrcaljenje slike i mijenjanje osvjetljenja slike. Šum se slici dodaje zbrajanjem matrice nasumično generiranih brojeva s originalnom matricom odnosno slikom. Vrijednosti koje su elementi generirane matrice mogli poprimiti su između nula i pet posto maksimalne moguće vrijednosti originalne matrice. U svrhe ovog rada ta vrijednost je 0.05 radi normalizacije podataka kako bi se poboljšala sposobnost učenja modela [38].

Rotiranje slike se postiže pomoću afinih transformacija (*engl. Affine Transformation*). Svaka slika se rotira između  $[0, 2\pi]$ . Zrcaljenje slike zrcali sliku preko njene vertikalne osi. Dešava svakoj slici s vjerojatnošću od pedeset posto. Zadnja korištena augmentacijska metoda je mijenjanje osvjetljenja. Ona se postiže množenjem matrice slike s nasumičnom vrijednošću između  $[0.9, 1.1]$ .

Oblik slike je broj redova i stupaca koje ona sadrži. Pošto se rad bavi frakturama oblika slika nije ujednačen. Vrijednosti stupaca i redaka su bile  $[70, 480]$  s većinom oko 128 zbog čega je veličina slike normalizirana na  $[128, 128]$ . Miješanje podataka se postiže nasumičnim miješanjem (*engl. Random Shuffle*) koji svaki podatak stavi na nasumičnu poziciju u nizu.

Parametri koji mijenjaju model su: korištena umjetna neuronska mreža, stopa učenja (*engl. Learning Rate*), funkcija greške (*engl. Loss*), optimizacijska funkcija (*engl. Optimizer*), veličina serije (*engl. Batch Size*), epoha treniranja (*engl. Epoch*), učestalost validacije (*engl. Valid Step*) i rano zaustavljanje (*engl. Early Stopping*).

Sve neuronske mreže objašnjene u poglavlju 2.3.1 su jedna od opcija za hiperparametar neuronska mreža. Stopa učenja je stopa koja određuje veličinu promjene u svakoj iteraciji modela. Što je stopa učenja veća to će svaka iteracija imati veći utjecaj na model i obratno. Isprobane stope učenja u FE modelima su  $\alpha \in \{0.0001, 0.00001, 0.000001\}$ .

Funkcija greške je metoda pomoću koje se ocjenjuje sposobnost neuronske mreže da modelira podatke za treniranje. Tijekom treniranja, cilj neuronske mreže je da se izlazna vrijednost funkcije greške smanji te time da se performanse neuronske mreže u modeliranju podataka poboljšaju. Funkcija greške korištena u ekstraktoru značajki

## Poglavlje 2. Definicija problema i metodologija

je binarna unakrsna entropija (*engl. Binary Cross-Entropy, BCE*). BCE je prosjek logaritma ispravljenih predviđenih vjerojatnosti i računa se pomoću formule 2.1.

$$BCELoss = -\frac{1}{N} \sum_{i=1}^N [y_i \cdot \log y'_i + (1 - y_i) \cdot \log (1 - y'_i)] \quad (2.1)$$

U formuli 2.1  $N$  predstavlja veličinu izlaza,  $y_i$  predstavlja ciljanu vrijednost, a  $y'_i$  modelom predviđenu vrijednost. Iz formule je očigledno da će BCE biti manji što su ciljana i predviđena vrijednost sličnije. Kao ciljanu vrijednost u FE koristila se slika frakture, dok se kao predviđenu vrijednost koristila rekonstruirana slika frakture.

Optimizacijske funkcije su algoritmi pomoću kojih smanjujemo funkciju greške. Ovisni su o parametrima pomoću kojih model strojnog učenja uči (na primjer težine neuronske mreže). Pomoću njih mijenjaju se navedeni parametri a time i model strojnog učenja. Optimizacijska funkcija korištena u ovom radu je *Adam* [39].

Veličina serije podrazumijeva broj podataka koje model strojnog učenja koristi tijekom jedne iteracije treniranja. Generalno, veća veličina serije znači stabilnije korake tijekom učenja do određene mjere – ovisno o problemu koji se rješava. Korištena vrijednost veličine serije je 32. Epoha treniranja je broj puta koje treniranje mora proći sve podatke za treniranje prije nego mora prestati s treniranjem. Granica ovog rada je bila 200 epoha treniranja.

Učestalost validacije je koliko često se model strojnog učenja tijekom treniranja testira na validacijskim podacima, to jest koliko često se poziva funkcija greške na validacijskim podacima. Učestalost validacije tijekom treniranja FE modela je jednom nakon svake epohe treniranja.

Zadnji hiperparametar je parametar koji definira rano zaustavljanje [40]. Njime se osigurava da model strojnog učenja prestane s treniranjem u trenutku kada se počne previše prilagođavati podacima za treniranje (*engl. Overfitting*) te time pogoršavati svoju generalizaciju nad drugim podacima. To se provjerava korištenjem funkcije greške na podacima koje model strojnog učenja ne koristi za treniranje, odnosno nad validacijskim podacima. Ako se vrijednost funkcije greške nad validacijskim podacima ne poboljša određen broj puta treniranje se zaustavlja. Taj broj je bio definiran hiperparametrom ranog zaustavljanja s vrijednošću od 10 iteracija funkcijom greške kroz validacijske podatke.

### 2.3.2 Podudaranje značajki

U ovom potpoglavlju objasniti će se komponente modela podudaranja značajki (*engl. Feature Matching, FM*) te algoritmi korišteni u njemu. FM implementira CIR način rada te je njegova svrha računanje sličnosti među rendgenskim slikama fraktura kako bi se mogle naći najslabije frakture između promatrane frakture i baze podataka. Potpoglavlje je podijeljeno na: *Korištene FE* gdje se opisuju načini korištenja treniranih FE modela, *Model sličnosti* gdje će DNN model korišten za stvaranje reprezentativnog vektora (*engl. Embedding*) biti objašnjen, *Korišteni hiperparametri* te *Funkcija greške* koja je izdvojena iz hiperparametara radi opsežnosti komponente.

Podatci u podudaranju značajki su podijeljeni na podatke za treniranje, podatke za validaciju i podatke za testiranje. Za razliku od FE, u FM će se testirati puno više modela te, pošto se performanse modela ocjenjuje pomoću validacijskih podataka, ne može se ignorirati pristranost koju će model indirektno stvoriti prema tim podacima. Jedini način da se garantiraju performanse modela je da se uvede još jedan skup podataka koje model neće vidjeti do samog kraja svog treniranja kada će biti iskorišteni za testiranje performansi modela i njegove sposobnosti generalizacije. Podijeljeni podatci su u omjeru 65% za treniranje, 20% za validaciju i 15% za testiranje odnosno 16650 za treniranje, 225 za validaciju i 171 za testiranje. Pošto podatci nisu balansirani, validacijski i testni podatci su limitirani na 20% odnosno 15% najmanje klase kako bi se osigurala balansiranoost podataka tijekom provjere performansi modela.

Međutim, pošto su podatci za treniranje ne balansirani primijenjena je metoda prekomjernog uzorkovanja (*engl. Oversampling*). Metoda prekomjernog uzorkovanja je kopiranje podataka koji su u manjini dok njihova količina ne pređe određeni broj. Taj broj je eksperimentalno postavljen na 1000.

#### Testirani FE modeli

Svaki istrenirani FE je upotrebljen u svrhe generiranja nove baze podataka kako bi se ubrzalo treniranje modela sličnosti u FM. Pošto su reprezentativni vektori iz FE u rasponu od nekoliko desetaka do stotina tisuća elemenata, primijenjena je tehnika analize glavnih komponenta (*engl. Principal Component Analysis, PCA*) koja služi



## Poglavlje 2. Definicija problema i metodologija

smanjivanju dimenzionalnosti vektora.

PCA funkcionira izračunavanjem matrica kovarijanci (*engl. Covariance Matrix*) podataka i njihovih svojstvenih vektora i svojstvenih vrijednosti (*engl. Eigenvectors and Eigenvalues*) te zadržavanjem samo onih s najviše informacija do određenog broja komponenti. Ako podatak ima malen broj svojstvenih vektora i vrijednosti algoritmu PCA se određuje broj komponenti koje može koristiti za opisivanje prostora podataka s fiksnim brojem umjesto postotkom informacija koje da zadrži. Broj komponentata podatka nakon PCA je ili broj komponentata koje sadrže barem 95% informacija podatka ili iz skupa brojeva [512, 1024, 2048, 4096] ako PCA može 99% informacija opisati s manje od 512 komponenti. Za određivanje PCA modela su upotrebljeni samo podatci za treniranje kako bi se očuvala nezavisnost performansi modela o validacijskim i test podacima.

### Model sličnosti

Za generiranje modela sličnosti za FM implementirana je posebna klasa koja omogućuje modularno stvaranje umjetnih neuronskih mreža. Navedenoj klasi se prosljeđuje niz slojeva koji predstavljaju kompoziciju modela te nizovi parametara potrebnih za inicijaliziranje komponenti modela. Moguće komponente modela sličnosti su: jednodimenzionalni konvolucijski sloj (*engl. One Dimensional Convolutional Layer, CNN*), jednodimenzionalni *average pooling* sloj (AVG), sloj spljoštavanja (*engl. Flatten Layer*), normalizirana eksponencijalna aktivacijska funkcija (*engl. Normalized Exponential Function, softmax*), ispravljena linearna aktivacijska jedinica (*engl. Rectified Linear Activation Unit, ReLu*), sloj ispadanja (*engl. Drop Layer*), sloj sažimanja (*engl. Squeeze Layer*), jednodimenzionalna normalizacija serije (*engl. One Dimensional Batch Normalization*) i potpuno povezana umjetna neuronska mreža (*engl. Fully Connected Artificial Neural Network, FC*).

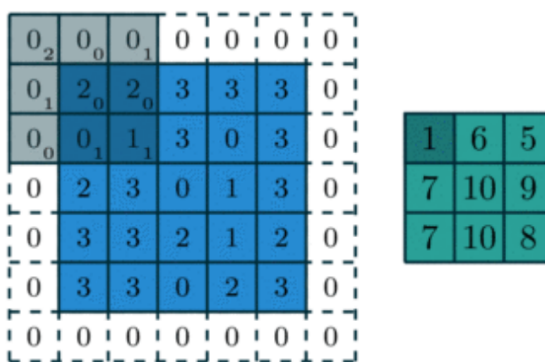
Slojevi spljoštavanja i sažimanja služe sličnoj svrsi, smanjivanju oblika (*engl. Shape*) podatka. Međutim, dok sloj sažimanja samo uklanja jednu dimenziju (npr. [32, 1, 128] → [32, 128]) sloj sploštavanja spoji dvije dimenzije i njihov sadržaj u jednu (npr. [32, 8, 126] → [32, 1008]). Stoga slojevi spljoštavanja i sažimanja imaju različite primjene. Još jedan sloj koji obrađuje podatke je sloj jednodimenzionalne

## Poglavlje 2. Definicija problema i metodologija

normalizacije serije. Namijenjen je za uporabu nad jednodimenzionalnim podacima. Koristi se nakon svakog sloja potpuno povezane umjetne neuronske mreže osim zadnjeg u svrhu normalizacije podataka.

Kako bi se poboljšale performanse umjetne neuronske mreže uz normalizaciju podataka koristi se i sloj ispadanja. Svrha sloja ispadanja je regularizacija neuronske mreže. Svaku epohu, sloj ispadanja predodređeni postotak nasumičnih težina sloja neuronske mreže postavi na nulu. Time neke dijelove mreže čini nevidljivim idućem neuronskom sloju, zbog čega će idući neuronski sloj naučiti temeljit svoj proračun na većem broju značajki.

Nakon pripreme podataka i težina slijede slojevi umjetnih neuronskih mreža. U te slojeve ubrajamo konvolucijske slojeve i sloj potpuno povezane umjetne neuronske mreže. Konvolucija [41, 42, 43] je matematička operacija množenja dvije matrice. Jedna matrica je podatak iz koje želimo detektirati značajke dok s drugom manjom matricom, koju zovemo *kernel*, vršimo operaciju matričnog množenja. Primjer konvolucije može se vidjeti na slici 2.7.



Slika 2.7 Primjer konvolucije, preuzeto s [44].

U CNN i AVG slojevima vrši se tip konvolucije. Sloj CNN sadrži težine kao svoje vrijednosti kernela koje se mijenjaju tijekom treniranja kako bi se izračunale što bolje značajke iz podatka. AVG s druge strane računa prosjek vrijednosti obuhvaćenih tijekom množenja s kernelom. FC je tip arhitekture umjetnih neuronskih mreža u kojem su svi neuroni jednog sloja povezani sa svim neuronima drugog sloja. Koriste se za obradu informacija i donošenje odluka.

## Poglavlje 2. Definicija problema i metodologija

Na izlazu svakog sloja umjetne neuronske mreže nalazi se odgovarajuća aktivacijska funkcija. Aktivacijska funkcija transformira izlaz sloja umjetne neuronske mreže te korištenjem matematičkih formula modificira vrijednosti izlaza. Ovisno o sloju korištena je jedna od dvije aktivacijske funkcije: ReLu i softmax. ReLu limitira izlaz neuronskog sloja na pozitivne vrijednosti. ReLu aktivacijska funkcija je prikazana u formuli 2.2. Softmax s druge strane (formula 2.3) na ulaz prihvata vektor od  $J$  pravih brojeva i generira distribuciju vjerojatnosti.

$$ReLU = x^+ = \max(0, x) \quad (2.2)$$

$$Softmax = \frac{e^{x_i}}{\sum_{j=1}^J e^{x_j}} \quad \text{for } i = 1, \dots, J \quad (2.3)$$

### Korišteni hiperparametri

U FM mogući hiperparametri su: FE (poglavlje 2.3.1), PCA (poglavlje 2.3.2), model sličnosti (poglavlje 2.3.2), metoda prekomjernog uzorkovanja, veličina serije, epoha treniranja, stopa učenja, optimizator, učestalost validacije, rano zaustavljanje, metode miješanja podataka i funkcija greške (poglavlje 2.3.1).

Veličina serije je postavljena na 64. Za razliku od FE ulazni podatci FM su veće kompleksnosti. Pošto je problematika traženja sličnih fraktura kompliciranija od rekonstrukcije slike, broj epoha treniranja je povećan s 200 na 300. Stopa učenja je postavljena na  $\alpha = 0.00001$  pošto je navedena stopa učenja dala najbolje rezultate kod treniranja FE. Koristio se isti optimizator kao u FE to jest *Adam* [39].

Učestalost validacije je smanjena na svaku treću seriju radi povećanja kompleksnosti problema. Ideja iza povećanja je ubrzavanje treniranja smanjivanjem nepotrebne validacije pošto je jedna serija premalo za znatne promjene u validacijskim podacima. Broj validacija nakon kojih bi se aktiviralo rano zaustavljanje modela je 20. Korištene funkcije greške su unakrsna entropija (*engl. Cross Entropy*) i funkcija gubitka trojki (*engl. Triplet*).

## Funkcija greške

Testirana su dva načina treniranja FE modela, svaki od kojih ima svoju funkciju greške i svoju metodu miješanja podataka (*engl. Shuffle Method*):

- Klasifikacijski model (*engl. Classification Model*)
- Model trojke (*engl. Triplet Model*)

Klasifikacijski model [6, 25, 5] je treniran pomoću klasificiranja podataka. Koristeći aktivacijsku funkciju *softmax* računaju se vjerojatnosti klasa koje se prosljede funkciji greške unakrsna entropija. Unakrsna entropija funkcionira na sličnom principu kao binarna unakrsna entropija no za razliku od binarne verzije, namijenjena je za više od jedne klase. Razlika je prikazana u formuli 2.4.

$$l(x, y) = L = \{l_1, \dots, l_N\}^T, \quad l_n = -w_{y_n} \log \frac{\exp(x_{n,y_n})}{\sum_{c=1}^C \exp(x_{n,c})} \cdot 1\{y_n \neq \text{ignore\_index}\} \quad (2.4)$$

U formuli 2.4  $w$  predstavlja težinu klase,  $C$  broj klasa,  $x$  ulaz, a  $y$  ciljanu klasu. Očigledno je da za razliku od BCE (formula 2.1) kod unakrsne entropije uračunat je utjecaj drugih klasa. Što je vjerojatnost krivih klasa veća to će vrijednost funkcije greške biti veća i obratno. Korištena metoda miješanja podataka je nasumično miješanje objašnjeno u FE (poglavlje 2.3.1).

Nakon što je klasifikacijski model istreniran, pretposljednji sloj je uzet kao vektor za izračun sličnosti između podataka. Za usporedbu sličnosti korištena je matematička funkcija Euklidske udaljenosti [22]. Euklidska udaljenost  $d_{x,y}$  se računa kao korijen kvadrata razlike dviju vrijednosti.

$$d_{x,y} = \sqrt{(x - y)^2} \quad (2.5)$$

U formuli 2.5  $x$  predstavlja reprezentativni vektor promatrane frakture, a  $y$  reprezentativni vektor frakture s kojom se promatrana fraktura uspoređuje.

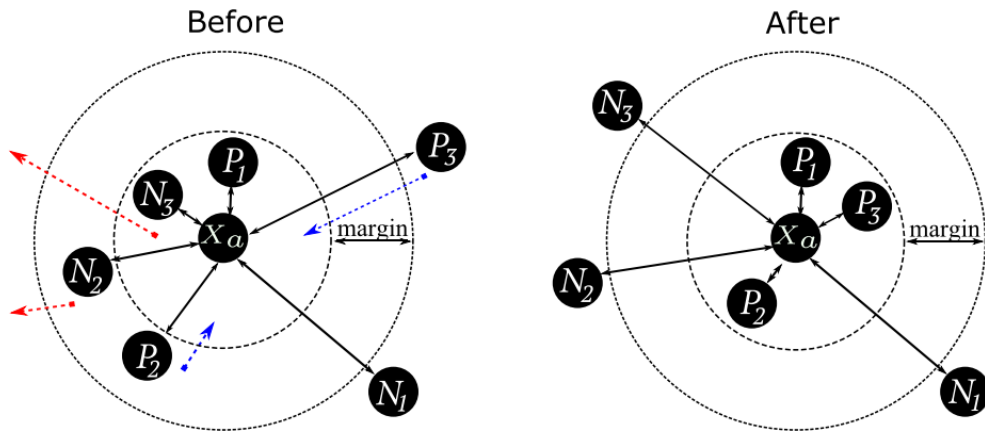
Za razliku od klasifikacijskog modela koji rješava problem klasifikacije, model trojke [45, 46, 5] uči međuklasnu i unutar klasnu udaljenost. Temelji se na funkciji gubitka trojki  $L_{a,p,n}$  koja se sastoji od 3 dijela: sidra (*engl. Anchor*), pozitivnog primjerka i negativnog primjerka.

Poglavlje 2. Definicija problema i metodologija

Sidro je neki primjerak koji će biti korišten kao referentna točka. Pozitivni primjerak je jedinka iste klase, dok negativni primjerak je jedinka različite klase od sidra. Oduzimanjem udaljenosti sidra i negativnog primjerka od udaljenosti sidra i pozitivnog primjerka može se vidjeti sposobnost računanja međuklasne i unutar klasne udaljenosti modela. Međutim, ako su pozitivni i negativni primjerci blizu, funkcija gubitka trojki će vratiti malu vrijednost. Stoga ubačena je i margina u formulu koja se zbraja s udaljenosti pozitivnog primjerka i sidra. Time se model uči da međuklasna udaljenost bude barem za marginu veća od unutar klasne udaljenosti. Opisana funkcija gubitka trojki prikazana je jednadžbom 2.6.

$$L_{a,p,n} = \frac{1}{N} \sum_i^N [f(x_i^a, x_i^p) - f(x_i^a, x_i^n) + m] \quad (2.6)$$

$N$  predstavlja broj trojki to jest kombinacija sidra, pozitivnih i negativnih primjeraka korištenih u seriji. Oznake  $x_i^a$ ,  $x_i^p$  i  $x_i^n$  predstavljaju  $i$ -tu kombinaciju sidra, pozitivne i negativne jedinice,  $f$  funkciju udaljenosti te  $m$  predstavlja marginu. Utjecaj funkcije gubitka trojki može se vidjeti na slici 2.8.



Slika 2.8 Primjer funkcije gubitka trojki, preuzeto sa [45]. Lijevo su prikazana tri para pozitivnih i negativnih primjeraka prije učenja pomoću funkcije gubitka trojki.  $P_i$  i  $N_i$  predstavljaju pozitivni i negativni par primjeraka u trojci dok je  $x_a$  sidro trojke. Desno je prikazana ciljana situacija udaljenosti među primjercima nakon učenja s funkcijom gubitka trojki.

Odabir trojki (*engl. Triplet Selection*) koji će se koristiti kao sidro te pozitivni

## Poglavlje 2. Definicija problema i metodologija

i negativni primjerak je od izrazite važnosti u modelu trojki. Ako se koriste samo bliski pozitivni i daleki negativni primjerci model će ili učiti jako sporo ili neće naučiti optimalnu reprezentaciju. Umjesto toga, nužno je odabrati trojke koje su modelu problematične. Korišteni odabir trojki u ovom radu je poluograničavajuća selekcija trojki (*engl. Semi Hard Triplet Selection*) [47, 46, 45]. Poluograničavajuća selekcija trojki bira sve trojke za koje vrijede ograničenja dana formulama 2.7 i 2.8.

$$f(x_i^a, x_i^p) < f(x_i^a, x_i^n) \quad (2.7)$$

$$f(x_i^a, x_i^p) + m > f(x_i^a, x_i^n) \quad (2.8)$$

Za razliku od klasifikacijskog modela, u modelu trojke udaljenost se računa pomoću formule 2.9.

$$izlaz[i, j] = (\text{podatak}[i, :] - \text{podatak}[j, :])^2 \quad (2.9)$$

S obzirom na mehaniku modela trojke, metoda miješanja podataka čini veliku razliku. Ako u iteraciji funkcije gubitka trojki postoji samo jedna jedinka neke klase ta klasa neće moći biti korištena kao sidro. Stoga, implementirana je posebna metoda miješanja podataka koja garantira da će u svakoj iteraciji funkcije gubitka trojki postojati barem dva primjerka svake klase ili niti jedan. Jedina moguća iznimka koja se može desiti ovoj metodi je ako zadnja iteracija sadrži malen broj primjeraka.

Kombiniranjem svih opisanih komponenti, razvijena metoda slijedi iduću listu koraka:

1. Učitaj promatranu sliku i izdvoji njene frakture
2. Ekstrahiraj značajke fraktura pomoću rekonstrukcijskog autoenkodera
3. Ekstrahirane značajke pretvori u reprezentativni vektor korištenjem modela sličnosti
4. Usporedi sličnost reprezentativnog vektora promatrane slike s reprezentativnim vektorima fraktura u bazi podataka korištenjem Euklidske udaljenosti
5. Sortiraj reprezentativne vektore baze podataka po sličnosti s reprezentativnim vektorom promatrane slike

# Poglavlje 3

## Rezultati

U ovom poglavlju bit će objašnjenje metrike korištene za mjerenje performansi FE u poglavlju 3.1.1 i FM u poglavlju 3.1.2 modela. Potom će rezultati istraživanja biti prezentirani u poglavlju 3.2.

### 3.1 Metrike

Metrike su vrijednosti koje iskazuju mjere performansi modela u nekoj problematici. Njima se prate i mjere sposobnosti modela tokom i nakon treniranja. U ovom radu koristile su različite metrike za FE i FM.

#### 3.1.1 Ekstraktor značajki, FE

Pošto FE rješava problem rekonstrukcije slike, odabrana je metrika Sørensen–Dice koeficijent (*engl. Sørensen–Dice Coefficient*). Sørensen–Dice koeficijent  $DSC$  je korišten u statistici za opisivanje sličnosti dvije jedinke. Računa se kao omjer dvostrukog presjeka između dvije jedinke i zbroja jedinki (formula 3.1).

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \quad (3.1)$$

$$DSC = \frac{2TP}{2TP + FP + FN} \quad (3.2)$$

### Poglavlje 3. Rezultati

U formuli 3.1  $X$  i  $Y$  predstavljaju jedinke koje se uspoređuju. U ovom radu navedene jedinke su originalna slika i rekonstruirana slika. Korištenjem Booleove algebre (*engl. Boolean Algebra*) dobije se formula 3.2. Varijable korištene u formuli  $TP$ ,  $FP$  i  $FN$  označavaju stvarno pozitivni (*engl. True Positive, TP*), lažno pozitivni (*engl. False Positive, FP*) i lažno negativni (*engl. False Negative, FN*) označeni podatci. U rekonstrukciji slike opisane vrijednosti se odnose na razlike u pikselima.

- Stvarno pozitivni - Kada je razlika između dva piksela malena (empirijski se uzela vrijednost tolerancije od  $|2.5\%|$ ) i originalna slika na tom pikselu ima vrijednost veću od nula
- Lažno pozitivni - Kada je razlika između dva piksela velika i taj piksel ima vrijednost veću od nula
- Lažno negativni - Kada je razlika između dva piksela velika i originalna vrijednost piksela je nula

#### 3.1.2 Podudaranje značajki

Za podudaranje značajki su istrenirana dva modela: klasifikacijski model i model trojke. Za klasifikacijski model iskorištena je metrika mjera F1 (*engl. F1 Score*) i točnost (*engl. Accuracy*), dok su za model trojke korištene metrike točnost, srednja kvadratna pogreška (*engl. Mean Squared Error, MSE*) i kosinusova sličnost (*engl. Cosine Similarity*).

#### Klasifikacijski model

Mjera F1 je prikaz modelove sposobnosti predviđanja pozitivnih jedinki. Računa se pomoću preciznosti (*engl. Precision*) i odziv (*engl. Recall*). Preciznost  $PPV$  je metrika koja označava koliko od svih, modelu pozitivnih jedinki, je stvarno pozitivno. To jest, ako model tvrdi za neku jedinku da je pozitivna koje su šanse da je stvarno pozitivna. Računa se pomoću formule 3.3.

$$PPV = \frac{TP}{TP + FP} \quad (3.3)$$



### Poglavlje 3. Rezultati

Odziv  $TPR$  je metrika koja za razliku od preciznosti govori o točnosti modela kod označavanja svih pozitivnih jedinki kao pozitivne (formula 3.4).

$$TPR = \frac{TP}{TP + FN} \quad (3.4)$$

Kako bi se dobila mjera  $F1$   $F1$  dvostruki umnožak preciznosti i odziv je podijeljen sa zbrojem preciznosti i odziv (formula 3.5).

$$F1 = \frac{2PPV \cdot TPR}{PPV + TPR} \quad (3.5)$$

Točnost  $T$  u problemu klasifikacije označava koliko često model točno odredi klasu jedinke. Računa se dijeljenjem točnih pogodaka sa sveukupnim brojem pogađanja (formula 3.6).

$$T = \frac{\text{broj točnih klasifikacija}}{\text{sveukupni broj klasifikacija}} \quad (3.6)$$

#### Model trojke

Točnost  $T_N$  kod modela trojke predstavlja koliki postotak fraktura u  $N$  najbližih primjeraka je iste klase kao promatrana fraktura. Računa se na sličan način kao kod klasifikacijskog modela (formula 3.6) gdje je broj fraktura istih klasa u  $N$  najbližijih fraktura podijeljen s  $N$  (formula 3.7).

$$T_N = \frac{\sum_{i=1}^N Y_i}{N}, \quad Y_i = \begin{cases} 1, & \text{ako su frakture iste klase} \\ 0, & \text{inace} \end{cases} \quad (3.7)$$

Metrika MSE predstavlja razliku između dvije jedinke, što je broj veći to je proporcionalno veća i razlika jedinke. Korišten je kao pokazatelj greške modela kod računanja sličnosti fraktura. Računa se korjenovanjem kvadrata razlike dvije frakture (formula 3.8).

$$mse = \sqrt{(\text{promatrana fraktura} - \text{fraktura iz baze})^2} \quad (3.8)$$

Kosinusova sličnost  $\cos \theta$  je mjera sličnosti između dva niza brojeva. Kod računanja kosinusove sličnosti dva niza brojeva promatraju se kao vektore prostora unutarnjeg umnoška (*engl. Inner Product Space*) čija dimenzionalnost je jednaka normi vektora

(broju elemenata niza). Prostor unutarnjeg umnoška je prostor realnih ili kompleksnih brojeva s operacijom unutarnjeg umnoška. Izlaz operacije unutarnjeg umnoška je skalar koji predstavlja kosinus kuta između dva vektora, odnosno skalarni produkt dva vektora. Navedeni kut može biti u rasponu  $[-1, 1]$  gdje jedan predstavlja dva slična vektora, nula dva ortogonalna vektora, a minus jedan dva suprotna vektora. Stoga, što je izlaz unutarnjeg umnoška bliži jedan, to su dva vektora odnosno frakture sličnije. Računa se pomoću formule 3.9 gdje su  $x_i$  i  $y_i$  elementi vektora  $x$  i  $y$ .

$$\cos \theta = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \quad (3.9)$$

## 3.2 Rezultati

### 3.2.1 Ekstraktor značajki

Performanse rekonstrukcijskih modela na validacijskim podacima prikazani su u tablicama 3.1 i 3.2. Tablica 3.1 prikazuje performanse rekonstrukcijskih modela dok tablica 3.2 prikazuje performanse pretreniranih rekonstrukcijskih modela na podacima za treniranje i podacima za validaciju.

Kao što se može vidjeti u poglavlju 2.3.1 određeni hiperparametri nisu mijenjani već su isti u svim modelima. Ti parametri su: optimizator *Adam*, BCE funkcija greške, dimenzija slike od 128 x 128 piksela, veličina serije od 32, broj epoha treniranja od 200 i učestalost validacije jednom nakon svake epohe treniranja.

Hiperparametar rano zaustavljanje je razlika između pretreniranih i ne pretreniranih modela. Kod pretreniranih modela, model se zaustavio tek nakon što nije bilo nikakvog poboljšanja performansa u validacijskim podacima barem 10 epoha treniranja te se pretrenirao na podacima za treniranje. Za razliku od pretreniranih modela, rano zaustavljanje ne pretreniranih modela se dogodi nakon što se performansa modela nije poboljšala za  $\geq 0.1\%$  barem 10 epoha treniranja.

Opservacijom performansama dobivenih metrikom Sørensen–Dice koeficijent, očigledno je da jednostavniji modeli UNet i CNN Autoenkoder postižu bolje rezultate. Iz tablica 3.1 i 3.2 može se iščitati da u većini slučajeva stopa učenja  $\alpha = 0.00001$  daje najbolje rezultate. Također, može se vidjeti da pretrenirani modeli (tablica 3.2)

### Poglavlje 3. Rezultati

Model	stopa učenja	Sørensen–Dice koeficijent
UNet	0.000001	99%
UNet	0.00001	99.8%
UNet	0.0001	99.94%
AttentionUNet	0.000001	47.79%
AttentionUNet	0.00001	94.35%
AttentionUNet	0.0001	99.99%
CNN Autoenkoder	0.000001	47.38%
CNN Autoenkoder	0.00001	98.63%
UNet3+	0.000001	52.66%
UNet3+	0.00001	80.73%
UNet3+	0.0001	60.27%

Tablica 3.1 Performanse rekonstrukcijskih modela.

većinom imaju bolje performanse, osim kod AttentionUNet modela sa stopom učenja od 0.00001 gdje su lošije.

Primjeri rekonstrukcija rekonstrukcijskih modela vidljivi su na slikama 3.1 i 3.2. Kao što se može uočiti na slikama, svi modeli su uspješno rekonstruirali sliku bez znatnog gubljenja informacija. Među slikama 3.1 može se primijetiti veća oštrina na slici prije rekonstrukcije (slika 3.1a) nego na onim poslije. Razlog tome je što je slika 3.1a jedna od većih prije rekonstrukcije s veličinom od 309x346 dok slika poslije rekonstrukcije ima standardiziranu rezoluciju od 128x128. Unatoč razlici veličina, slika 3.1a je izabrana kao jedan od primjera radi količine detalja sadržanih na njoj.

Usporede li se naslovi rekonstrukcijskih modela prikazanih na slikama i tablicama, može se primijetiti da prikazani primjer AttentionUNet rekonstrukcije ne koristi najbolji model. To je napravljeno jer svi AttentionUNet modeli sa stopom učenja od 0.0001 unatoč dobrim performansama tijekom treniranja rekonstruira slike nalik na šum (slika 3.3).

### Poglavlje 3. Rezultati

Model	stopa učenja	Sørensen–Dice koeficijent
UNet	0.000001	99.99%
UNet	0.00001	99.99%
UNet	0.0001	99.99%
AttentionUNet	0.000001	73.67%
AttentionUNet	0.00001	78.9%
AttentionUNet	0.0001	99.99%
CNN Autoenkoder	0.000001	98.42%
CNN Autoenkoder	0.00001	99.83%
UNet3+	0.000001	91.23%
UNet3+	0.00001	92.18%
UNet3+	0.0001	70.16%

Tablica 3.2 Performanse pretreniranih rekonstrukcijskih modela.

#### 3.2.2 Podudaranje značajki

Prilikom mjerenja podudaranja značajki evaluirala su se tri parametra: način treniranja (klasifikacija, trojka) koji određuje metodu miješanja podataka i funkciju greške, arhitektura modela sličnosti i FE (izabrani rekonstrukcijski model).

Ostali parametri su konstantni i imaju vrijednosti: epoha treniranja je 300, veličina serije 64, učestalost validacije je jednom nakon svake treće epohe treniranja, stopa učenja  $\alpha = 0.00001$ , optimizator *Adam* s propadanjem težina od 0.00001, prekomjernim uzorkovanjem od tisuću i rano zaustavljanje nakon 20 validacija bez napredovanja validacijskih performansi.

Arhitektura modela sličnosti	oznaka arhitekture modela sličnosti
sloj ispadanja, FC, <i>ReLU</i> , normalizacija serije	FC1
sloj ispadanja, FC, normalizacija serije, <i>ReLU</i>	FC2
sloj ispadanja, FC, <i>ReLU</i>	FC
CNN, <i>ReLU</i> , sloj spljoštavanja	CNN
CNN, AVG, <i>ReLU</i> , sloj spljoštavanja	CNNAvg

Tablica 3.3 Oznake gradivnih blokova arhitekture modela sličnosti.

Isprobane arhitekture modela sličnosti i njihove oznake prikazane su u tablici 3.3.

Poglavlje 3. Rezultati



(a) Slika prije rekonstrukcijskog modela (b) UNet, stopa učenja  $\alpha = 0.0001$  (c) AttentionUNet, stopa učenja  $\alpha = 0.00001$



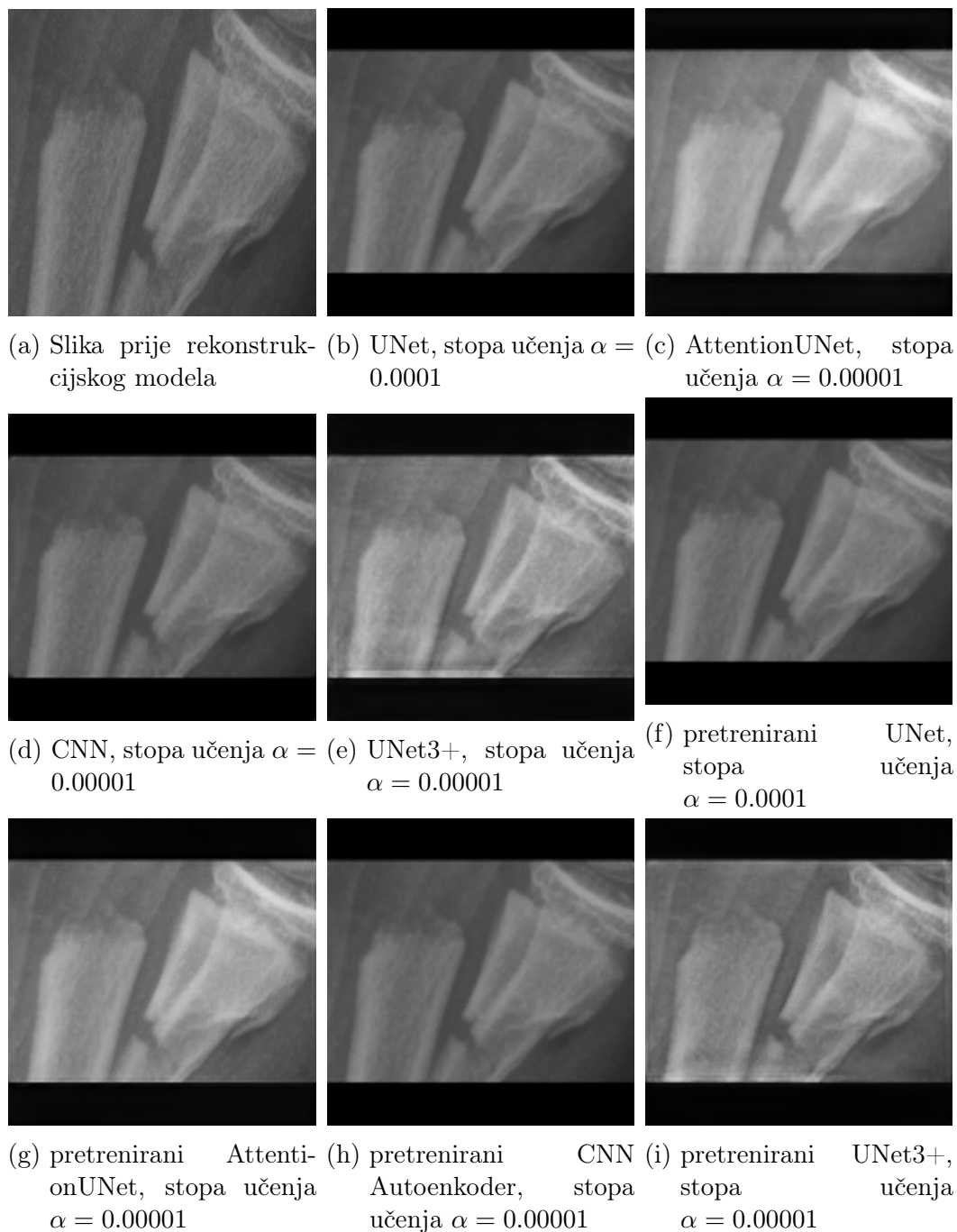
(d) CNN, stopa učenja  $\alpha = 0.00001$  (e) UNet3+, stopa učenja  $\alpha = 0.00001$  (f) pretrenirani UNet, stopa učenja  $\alpha = 0.0001$



(g) pretrenirani AttentionUNet, stopa učenja  $\alpha = 0.00001$  (h) pretrenirani CNN Autoenkoder, stopa učenja  $\alpha = 0.00001$  (i) pretrenirani UNet3+, stopa učenja  $\alpha = 0.00001$

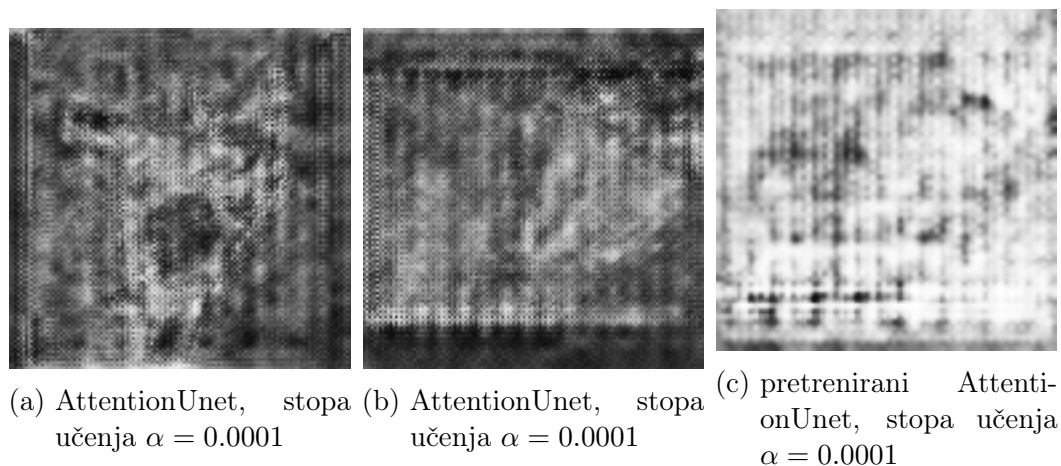
Slika 3.1 Primjeri rekonstrukcije slika s najboljim modelima iz tablica 3.1 i 3.2.

Poglavlje 3. Rezultati



Slika 3.2 Primjeri rekonstrukcije slika s najboljim modelima iz tablica 3.1 i 3.2.

### Poglavlje 3. Rezultati



Slika 3.3 Iznimke jednakosti najbolje rekonstrukcije i najboljeg Sørensen–Dice koeficijenta kod AttentionUnet modela.

U njoj se nalaze niz slojeva i definirana oznaka za taj niz respektivno. Svaki sloj u modelu sličnosti, osim posljednjeg, podrazumijeva jednu od arhitektura prikazanih u tablici 3.3. Posljednji sloj se sastoji od sloja ispadanja i FC sloja. U klasifikacijskom modelu dodan je još jedan FC sloj veličine broja klasa u bazi podataka.

Iz tablice 3.4 mogu se iščitati kombinacije generiranih PCA komponenti i FE modela korištenih za pretvorbu podataka u reprezentativne vektore. Broj komponenti PCA je određen ili brojem ili postotkom. U slučaju broja, broj komponenti iskorištenih za opis podataka bit će jednak broju prikazanom u tablici 3.4. Postoci zadržavaju onoliko komponenti koliko je potrebno da bi se u podatku nakon PCA nalazilo 95% informacija originalnog podatka. Imena modela nastala su spajanjem imena FE modela i brojem poslanim u PCA algoritam (npr. UNet 512 ili UNet3+ 0.95).

Tablica 3.5 pomoću oznaka iz tablice 3.3 prikazuje sve FC arhitekture koje su bile trenirane i njihove oznake. Oznaka isprobane arhitekture sastoji se od vrste FC arhitekture i broja ponavljanja te arhitekture. Uz arhitekture iz tablice 3.5 isprobane su i kombinacije CNN FC1\_4, CNN FC1\_5, CNNAvg FC1\_4, CNNAvg FC1\_5, NN FC2\_4, CNN FC12\_5, CNNAvg FC2\_4, CNNAvg FC2\_5.

U tablici 3.6 su prikazani brojevi neurona pojedinih slojeva te kombinacije ne-

### Poglavlje 3. Rezultati

FE	PCA
CNN Autoenkoder	512 / 1024 komponenti
UNet	512 / 1024 / 2048 komponenti
AttentionNnet	0.95% informacija
UNet3+	0.95% informacija

Tablica 3.4 Isprobane kombinacije FE i PCA.

Kombinacije arhitektura modela sličnosti	oznaka kombinacije
FC1, FC1, FC1	FC1_3
FC1, FC1, FC1, FC1	FC1_4
FC1, FC1, FC1, FC1, FC1	FC1_5
FC2, FC2, FC2	FC2_3
FC2, FC2, FC2, FC2	FC2_4
FC2, FC2, FC2, FC2, FC2	FC2_5
FC, FC, FC	FC_3

Tablica 3.5 Isprobane arhitekture modela sličnosti i njihove oznake.

urona koje su isprobane s tri, četiri i pet skrivenih FC slojeva. Prikazane su i oznake koje predstavljaju broj FC slojeva istreniranog modela te koja kombinacija neurona je korištena.

Koristeći do sad objašnjene tablice ako se želi prikazati AttentionNet model s PCA vrijednosti od 0.95% (tablica 3.4), tri ponavljanja FC1 arhitekture (tablica 3.5), i brojevima neurona u 3 skrivena sloja od [4096, 2048, 2048] (tablica 3.6) napisat će se AttentionNet 0.95 FC1\_3 3\_sloja\_2. Za UNet s PCA od 512 komponenti i ostalim hiperparametrima istim kao u prijašnjem primjeru, napisalo bi se UNet 512 FC1\_3 3\_sloja\_2.

### Klasifikacijski model

U tablici 3.7 prikazani su najbolji klasifikacijski FM modeli i njihove performanse. Svi modeli iz tablice 3.7 sastojali su se od tri skrivena sloja u svrhu pronalaženja najboljeg FE modela za klasifikacijsku problematiku.

Arhitekture istreniranih FM modela tablice 3.7 su prikazane kombinacijom oz-



Poglavlje 3. Rezultati

Brojevi neurona u skrivenim slojevima	oznaka kombinacije slojeva
[4096, 4096, 4096]	3_sloja_1
[4096, 2048, 2048]	3_sloja_2
[2048, 1024, 1024]	3_sloja_3
[4096, 4096, 4096, 4096]	4_sloja_1
[4096, 4096, 2048, 2048]	4_sloja_2
[4096, 2048, 1024, 1024]	4_sloja_3
[2048, 1024, 1024, 512]	4_sloja_4
[2048, 2048, 1024, 1024]	4_sloja_5
[2048, 1024, 1024, 1024]	4_sloja_6
[4096, 4096, 4096, 4096, 4096]	5_sloja_1
[4096, 4096, 4096, 2048, 2048]	5_sloja_2
[4096, 4096, 2048, 1024, 1024]	5_sloja_3
[4096, 2048, 1024, 1024, 1024]	5_sloja_4
[4096, 2048, 1024, 1024, 512]	5_sloja_5

Tablica 3.6 Brojevi neurona korišteni u skrivenim slojevima i oznaka kombinacije.

naka objašnjenih u tablicama 3.3, 3.4, 3.5 i 3.6. Iz tablice 3.7 može se iščitati da li su podatci prikazanih FM modela generirani pomoću pretreniranog FE modela ili ne. Također je zapisana metrika mjera F1 kao pokazatelj performansi klasifikacijskih modela.

U tablici 3.8 prikazani su najbolji AttentionUNet klasifikacijski modeli s tri, četiri ili pet skrivenih slojeva. Uspoređujući rezultate tablica 3.7 i 3.8 uočljivo je da su svi modeli došli do 40% mjere F1 s najvišim mjerama kod modela UNet3+ 0.95 FC2\_3 3\_sloja\_2 i modela AttentionUnet 0.95 FC2\_3 3\_sloja\_2 s pretreniranim FE. Također, vidi se da povećanje broja skrivenih slojeva ili neurona u skrivenim slojevima nije imao znatnih pozitivnih utjecaja na performanse klasifikacijskog AttentionUNet modela.

Na slikama 3.4 prikazane su matrice zabune klasifikacijskih modela UNet3+ 0.95 FC2\_3 3\_sloja\_2 i AttentionUnet 0.95 FC2\_3 3\_sloja\_2. Matrica zabune prikazuje odnos predviđanja modela za određene ulazne podatke i ispravnog predviđanja (ispravne oznake) za svaki od ulaznih podataka. Korišteni podatci u matricama zabune su podatci iz validacijskog i iz testnog skupa. Validacijski podatci su prikazani u lijevom stupcu, dok su testni podatci prikazani u desnom stupcu 3.4. Prvi red

### Poglavlje 3. Rezultati

FM model	pretrenirani FE	mjera F1
UNet 1024 FC1_3_3_sloja_2	ne	39.11%
CNN Autoenkoder 512_3 FC1_3_sloja_2	ne	42.67%
UNet3+ 0.95 FC2_3_3_sloja_2	ne	43.11%
AttentionUnet 0.95 FC2_3_3_sloja_3	ne	41.78%
UNet3+ 0.95 FC2_3_3_sloja_3	da	41.33%
AttentionUnet 0.95 FC2_3_3_sloja_2	da	43.56%

Tablica 3.7 Najbolje performanse klasifikacijskih modela s 3 skrivena sloja.

FM model	pretrenirani FE	mjera F1
AttentionUnet 0.95 FC2_3_3_sloja_2	ne	41.78%
AttentionUnet 0.95 FC2_4_4_sloja_4	ne	41.78%
AttentionUnet 0.95 FC1_5_5_sloja_5	ne	42.67%

Tablica 3.8 Najbolje performanse AttentionUNet klasifikacijskih modela s tri, četiri i pet skrivenih slojeva.

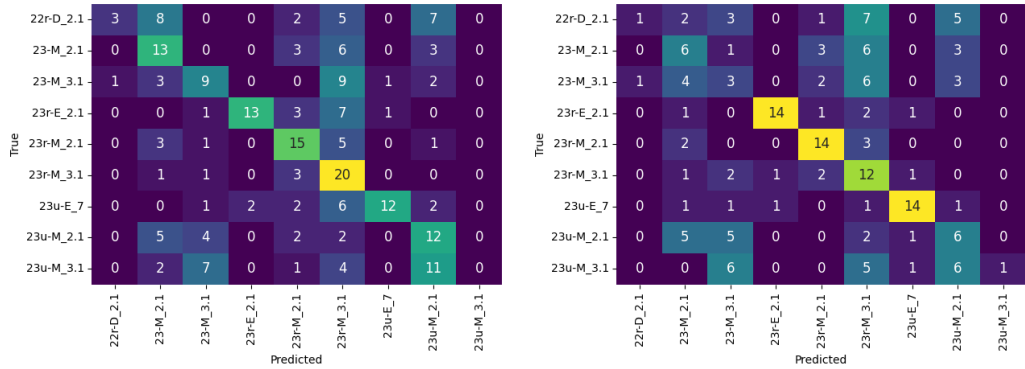
sadrži rezultate modela AttentionUnet 0.95 FC2\_3\_3\_sloja\_2 dok drugi red sadrži rezultate modela UNet3+ 0.95 FC2\_3\_3\_sloja\_2. Iz slike se može primijetiti kako oba modela imaju slične performanse i slične probleme.

Iz tablice 3.9 može se iščitati da je AttentionUnet 0.95 FC2\_3\_3\_sloja\_2 postigao bolje performanse od UNet3+ modela. Tablica 3.9 sadrži preciznost, odziv i mjeru F1 svake klase oba modela nad testnim podacima. Uz to, sadrži i prosjeke navedenih metrika i sveukupnu točnost modela. Klase s kojima su modeli imali najviše poteškoća, sortirane silazno po performansama, su 23-M\_3.1, 23u-M.3.1 i 22r-D\_2.1. Mogući razlozi loših performansi bit će diskutirani u poglavlju 4. AttentionUNet modelu su se performanse na testnim podacima smanjile za 1.61%, dok su UNet3+ modelu performanse manje za 6.34%. Iz toga se može zaključiti da AttentionUNet sadrži bolju sposobnost generalizacije od UNet3+ modela.

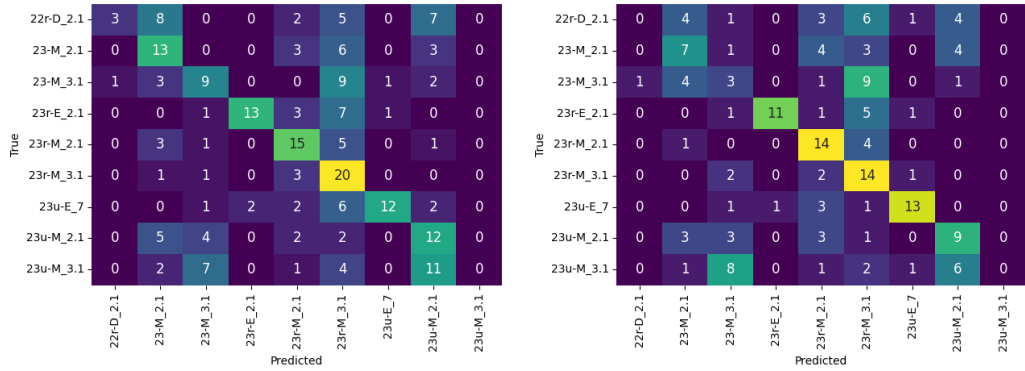
### Model trojke

Za razliku od klasifikacijskih modela, model trojke koristi točnost, MSE i kosinusovu sličnost kao metrike. U tablicama 3.10 i 3.11 su prikazani naziv FM model, da li je

### Poglavlje 3. Rezultati



(a) Matrica zabune modela AttentionU-net 0.95 FC2\_3 3\_sloja\_2 na validacijskim podacima (b) Matrica zabune modela AttentionU-net 0.95 FC2\_3 3\_sloja\_2 na testnim podacima



(c) Matrica zabune modela UNet3+ 0.95 FC2\_3 3\_sloja\_2 na validacijskim podacima (d) Matrica zabune modela UNet3+ 0.95 FC2\_3 3\_sloja\_2 na testnim podacima

Slika 3.4 Matrice zabune najboljih klasifikacijskih modela na validacijskim i testnim podacima.

FE model korišten za generiranje podataka pretreniran,  $T_{N=5}$  (formula 3.7) i  $T_{N=1}$  (formula 3.7). Točnost (formula 3.7) predstavlja postotak najbližih fraktura koje su iste klase kao promatrana fraktura.  $T_{N=5}$  (formula 3.7) je postotak frakture iste klase u najbližih pet fraktura dok je  $T_{N=1}$  (formula 3.7) postotak fraktura koje su iste klase kao najbliža fraktura.

Tablice performansi modela sličnosti su podijeljene na performanse modela trojke s tri skrivena sloja (tablica 3.10), gdje su prikazani najbolji modeli trojke za svaki

Poglavlje 3. Rezultati

	preciznost	odziv	mjera F1	preciznost	odziv	mjera F1
	UNet3+ 0.95 FC2_3 3_sloja_2			AttentionUnet 0.95 FC2_3 3_sloja_2		
22r-D_2.1	0	0	0	50%	5.26%	9.52%
23-M_2.1	35%	36.84%	35.9%	27.27%	31.58%	29.27%
23-M_3.1	15%	15.79%	15.38%	14.29%	15.79%	15%
23r-E_2.1	91.67%	57.89%	70.97%	87.5%	73.68	80%
23r-M_2.1	43.75%	73.68%	54.9%	60.87%	73.68%	66.67%
23r-M_3.1	31.11%	73.68%	43.75%	27.27%	63.16%	38.1%
23u-E_7	76.48%	68.42%	72.22%	77.78%	73.68	75.68%
23u-M_2.1	37.5%	47.37%	41.86%	25%	31.58%	27.9%
23u-M_3.1	0%	0%	0%	100%	5.26%	10%
prosjek	36.72%	41.52%	37.22%	52.22%	41.52%	39.13%
točnost	41.52%			41.52%		

Tablica 3.9 Performanse najboljih klasifikacijskih modela nad testnim podacima.

FE model, i performanse modela AttentionUNet trojke s tri četiri ili pet skrivenih slojeva koje su prikazane u tablici 3.11.

Iz tablice 3.10 može se primijetiti da kod modela trojke FE ima puno veći utjecaj na performanse nego u klasifikaciji. Najbolji model s tri skrivena sloja je ne pre-trenirani AttentionUnet 0.95 FC2\_3 3\_sloja\_2 s točnosti 1 od 75% i točnosti 5 od 48.67%. Drugi najbolji je UNet 2048 FC1\_3 3\_sloja\_2 s točnosti 1 od 70.83% i točnosti 5 od 45.13%.

Kod modela trojke povećanje skrivenih slojeva s tri na četiri je rezultiralo u poboljšanju performansi te je najbolji model trojke AttentionUnet 0.95 FC2\_4 4\_sloja\_1 (tablica 3.11). Međutim, ako se nastavilo s povećanjem broja slojeva, uslijed pre-tjerane prilagodbe modela podacima za treniranje, konačni rezultat je bio lošiji u odnosu na rezultat koji modeli postižu s manjim brojem slojeva.

Dodatno su testirani i CNN modeli s tri, četiri i pet skrivenih slojeva, no najbolji CNN model je AttentionUnet 0.95 CNN FC1\_5 5\_sloja\_2 s točnosti 5 od 29.16% i točnosti 1 od 33.33%.

### Poglavlje 3. Rezultati

FM model	pretrenirani FE	$T_{N=5}$	$T_{N=1}$
UNet 2048 FC1_3 3_sloja_2	ne	45.13%	70.83%
CNN Autoenkoder 1024 FC1_3 3_sloja_1	ne	40.68%	54.17%
UNet3+ 0.95 FC1_3 3_sloja_1	ne	43.36%	62.5%
UNet3+ 0.95 FC2_3 3_sloja_3	ne	48.25%	50%
AttentionUnet 0.95 FC2_3 3_sloja_2	ne	48.67%	75%
UNet3+ 0.95 FC2_3 3_sloja_3	da	44.92%	66.67%
AttentionUnet 0.95 FC1_3 3_sloja_2	da	44.16%	62.5%

Tablica 3.10 Performanse modela trojke s tri skrivena sloja.

FM model	pretrenirani FE	$T_{N=5}$	$T_{N=1}$
AttentionUnet 0.95 FC2_3 3_sloja_2	ne	48.67%	75%
AttentionUnet 0.95 FC2_4 4_sloja_1	ne	50.86%	79.17%
AttentionUnet 0.95 FC1_5 5_sloja_3	ne	43.48%	75%
AttentionUnet 0.95 CNN FC1_5 5_sloja_2	ne	29.16%	33.33%

Tablica 3.11 Performanse modela trojke AttentionUNet s tri, četiri ili pet skrivenih slojeva.

### Usporedba klasifikacijskog modela i modela trojke

Na slikama 3.5, 3.7 i 3.6 mogu se vidjeti primjeri najboljih FM modela iz modela trojki i klasifikacijskog modela. Slika u gornjem lijevom kutu predstavlja promatranu frakturu, odnosno frakturu kojoj tražimo najbližnje frakture. Pri vrhu svake pod rendgenske slike (na slikama 3.5, 3.7 i 3.6) napisana je klasa prikazane frakture te njena kosinusova sličnost s prvom to jest promatranom frakturom. Odabrana promatrana fraktura je iz testnog skupa podataka dok su sve ostale frakture iz skupa podataka za treniranje. Klasa odabrane promatrane frakture je klasa koja je najviše puta predviđena u matrici zabune (slika 3.4) sa svrhom demonstriranja performansi razlikovanja sličnih slika.

Temeljem vizualne inspekcije, svi modeli su uspjeli naći slike s visokom kosinusovom sličnošću i sličnim izgledom. Jedini model koji je uspio dobro odrediti među klasnu i unutar klasnu udaljenost je Net3+ 0.95 FC2\_3 3\_sloja\_2 (slika 3.6). Na slikama 3.6 se može vidjeti da su UNet3+ slike i promatrana slika (gornji lijevi kut)

### Poglavlje 3. Rezultati

sve iste klase, čineći njegovu  $T_{N=9} = 100\%$  (formula 3.7). U tablici 3.9 vidi se da je preciznost UNet3+ modela s klasom 23r-M\_3.1 samo 31.11%. Sa sličnom preciznošću AttentionUNet klasifikacijski model ne uspijeva naći ni jednu sliku iste klase unatoč sličnosti nađenih slika. AttentionUNet triplet daje slične rezultate.

U prilogu B su prikazane još dvije klase u dva različita ekstrema. Ekstrem male među klasne sličnosti prikazan je na slikama B.4, B.6 i B.5 na kojima se nalazi klasa "23r-E\_2.1". Klasa frakture "23r-E\_2.1" spada pod najrjeđu klasu (tablica 2.1) sa 106 primjeraka u sveukupnoj bazi podataka. Unatoč malom broju podataka, modeli su uspješni naučiti klasificirati ju s visokom točnošću 3.4. Na slikama B.4, B.6 i B.5 vidi se da su svi FM modeli vratili devet najbližijih fraktura iste klase kao promatrana fraktura s  $T_{N=9} = 100\%$  (formula 3.7). Od tri prikazana modela FM AttentionUNet 0.95 CNN FC1\_5 5\_sloja\_2 ima najveće vrijednosti kosinusove sličnosti. Vrijedi primijetiti kako je kosinusova sličnost AttentionUNet modela trojke za 10% veća u ovom primjeru nego kod primjera prikazanog na slici 3.5, odnosno AttentionUNet model je pouzdaniji model u odnosu na UNet3+.

Međutim, u drugoj krajnosti kod slučaja s pre velikom među klasnom sličnošću modeli imaju poteškoća u razdvajanju klasa kao što je prikazano na slikama B.1, B.2 i B.3. Iako prikazane rendgenske slike fraktura jesu izgledom slične, njihove prave klase nisu iste. Više o mogućim razlozima navedenog problema u idućem poglavlju 4.

U tablici 3.12 prikazana je  $T_{N=5}$  i  $T_{N=1}$  (formula 3.7), odnosno postotak najbližih fraktura koje su iste klase kao promatrana fraktura. Sve frakture iz testnih podataka su korištene kao promatrana fraktura te su najbliže frakture tražene u podacima za treniranje. Iz tablice 3.12 mogu se iščitati metrike korištene za procjenu sličnosti fraktura s promatranom frakturom i performanse istreniranih modela. Podatci su sortirani slizano po kosinusovoj sličnosti, a uzlazno po MSE. Prikazane su performanse modela trojke AttentionUnet 0.95 FC2\_4 4\_sloja\_1, klasifikacijskog modela AttentionUnet 0.95 FC2\_3 3\_sloja\_2 i klasifikacijskog modela UNet3+ 0.95 FC2\_3 3\_sloja\_2.

Usporede li se prikazani rezultati s validacijskim rezultatima tijekom treniranja (tablice 3.9 i 3.11) primijetit će se znatno pogoršanje. Razlog pada performansi je korištenje samo validacijskih podataka tijekom validacije čime se znatno smanjila

Poglavlje 3. Rezultati

Metrike	AttentionUnet trojka	AttentionUnet klasifikacija	UNet3+
$\cos T_{N=5}$ (formula 3.7)	37.94%	39.58%	39.91%
$\cos T_{N=1}$ (formula 3.7)	39.18%	36.84%	38.01%
MSE $T_{N=5}$ (formula 3.7)	34.87%	34.8%	35.5%
MSE $T_{N=1}$ (formula 3.7)	34.50%	33.33%	35.67%

Tablica 3.12 Točnost najboljih modela podudaranja slika koristeći metrike kosinusu sličnost i MSE.

raznolikost podataka u bazi pretrage i utjecaj potencijalno krivog anotiranja.

Kako bi se smanjio utjecaj grešaka u anotacijama, klase su spojene u natklase zbog čega se nova baza podataka sastoji od tri klase: E, M i D. U tablicama 3.13, 3.14 i 3.15 nalaze se performanse modela kod pronalaženja najsličnijih slika iste natklase. U lijevom stupcu se nalaze metrike, dok su u desnom model, performanse i natklase. Može se vidjeti znatno poboljšanje performansi svih modela. Jedini stupac s istim performansama je stupac klase D.

Metrike	UNet3+		
prosječna $\cos T_{N=5}$ (formula 3.7)	78.44		
prosječna MSE $T_{N=5}$ (formula 3.7)	72.73		
Natklasa	E	M	D
$\cos T_{N=5}$ (formula 3.7)	76.15%	90.95%	7.89%
MSE $T_{N=5}$ (formula 3.7)	69.08%	86.07%	0.0%

Tablica 3.13 Točnost klasifikacijskog modela UNet3+ 0.95 FC2\_3 3\_sloja\_2 u podudaranju natklasa slika koristeći metrike kosinusu sličnost i MSE.

Metrike	AttentionUnet klasifikacija		
prosječna $\cos T_{N=5}$ (formula 3.7)	79.09		
prosječna MSE $T_{N=5}$ (formula 3.7)	75.4		
Natklasa	E	M	D
$\cos T_{N=5}$ (formula 3.7)	82.73%	89.3%	10.53%
MSE $T_{N=5}$ (formula 3.7)	72.4%	88.93%	0.99%

Tablica 3.14 Točnost klasifikacijskog modela AttentionUnet 0.95 FC2\_3 3\_sloja\_2 u podudaranju natklasa slika koristeći metrike kosinusu sličnost i MSE.

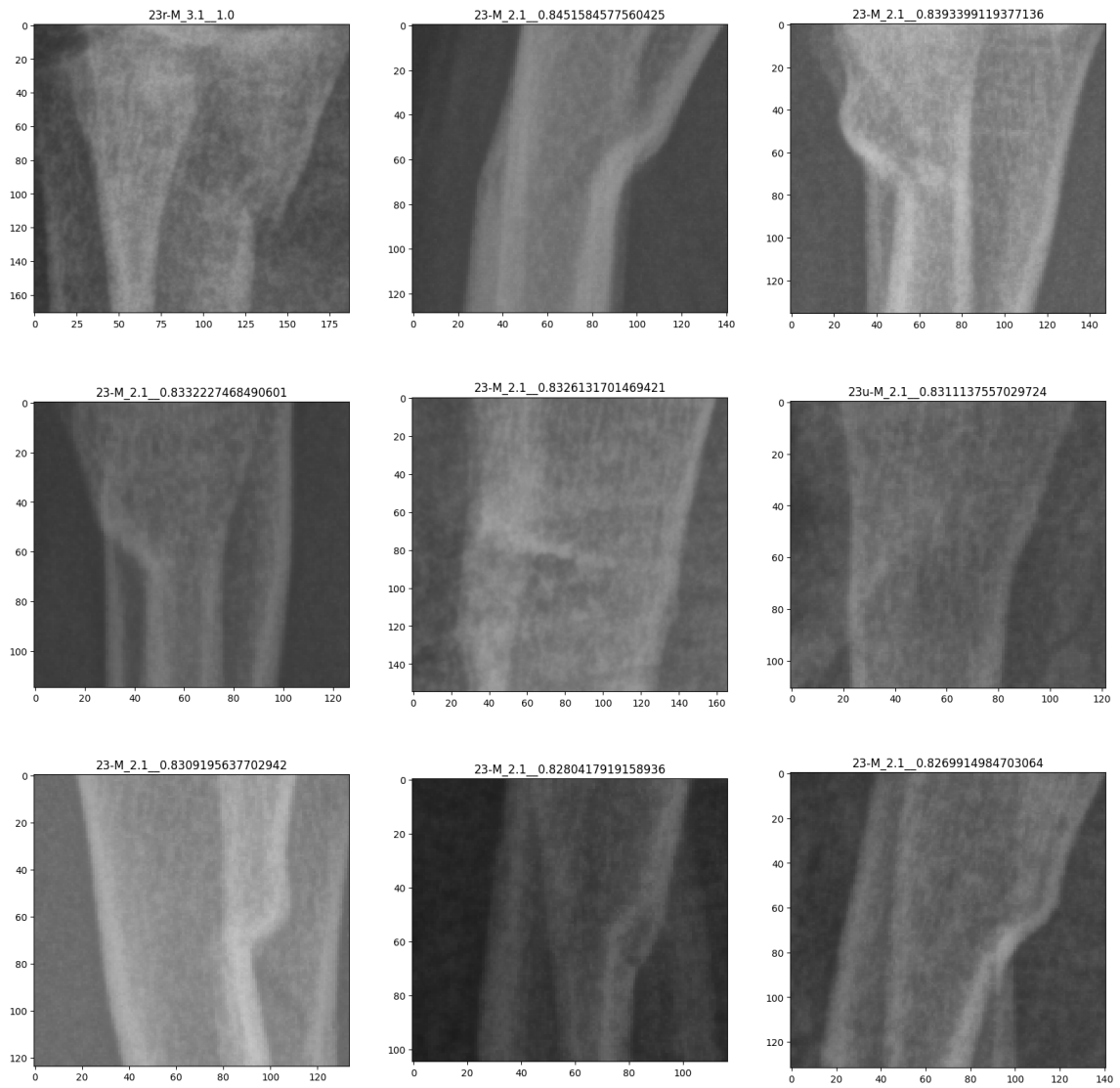
Poglavlje 3. Rezultati

Metrike	AttentionUnet trojka		
prosječna $\cos T_{N=5}$ (formula 3.7)	79.46		
prosječna MSE $T_{N=5}$ (formula 3.7)	78.73		
Natklasa	E	M	D
$\cos T_{N=5}$ (formula 3.7)	77.96%	92.11%	6.58%
MSE $T_{N=5}$ (formula 3.7)	68.42%	95.29%	0.0%

Tablica 3.15 Točnost modela trojke AttentionUnet 0.95 FC2\_4 4\_sloja\_1 u podudaranju natklasa slika koristeći metrike kosinusovu sličnost i MSE.

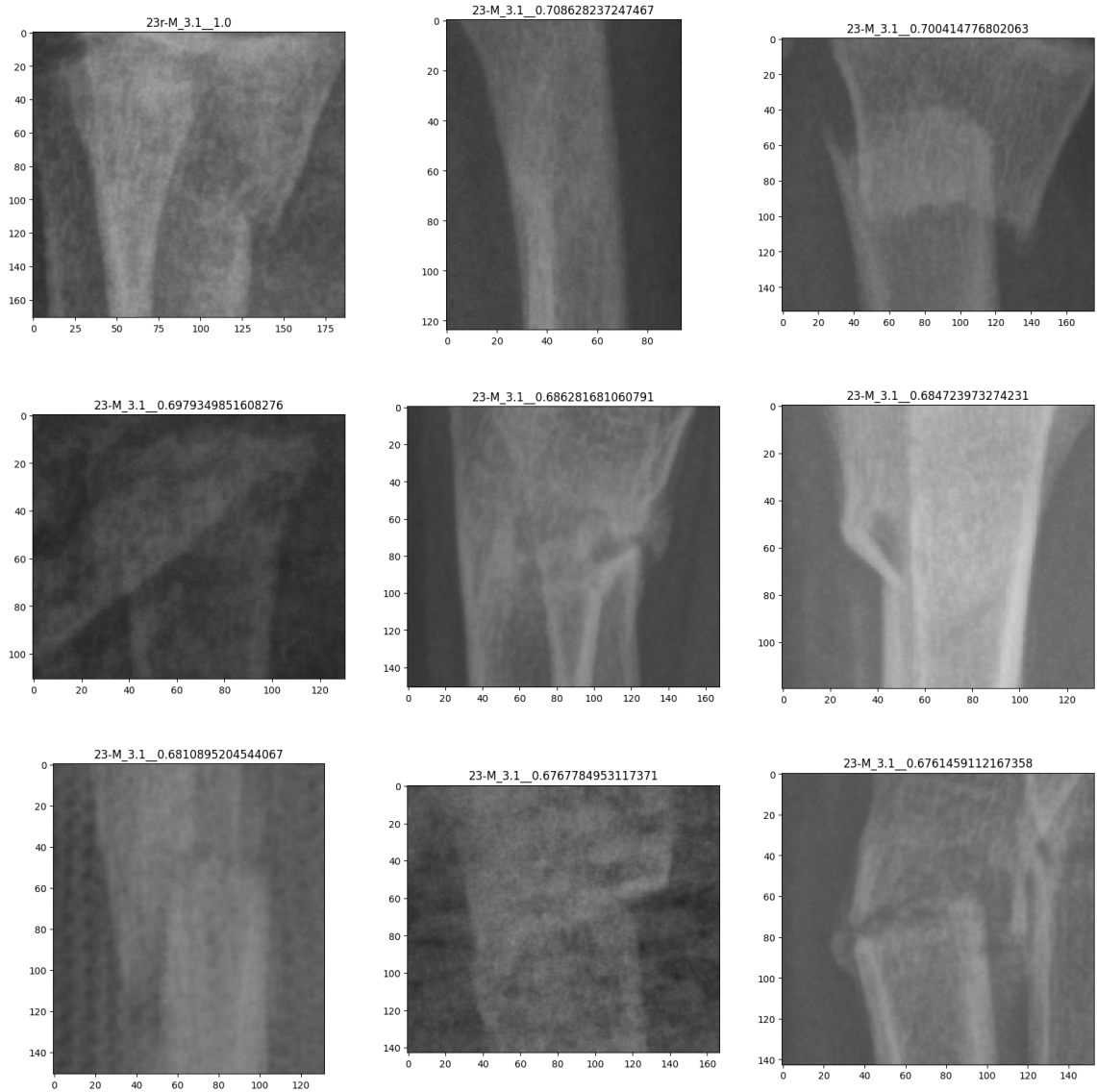


### Poglavlje 3. Rezultati



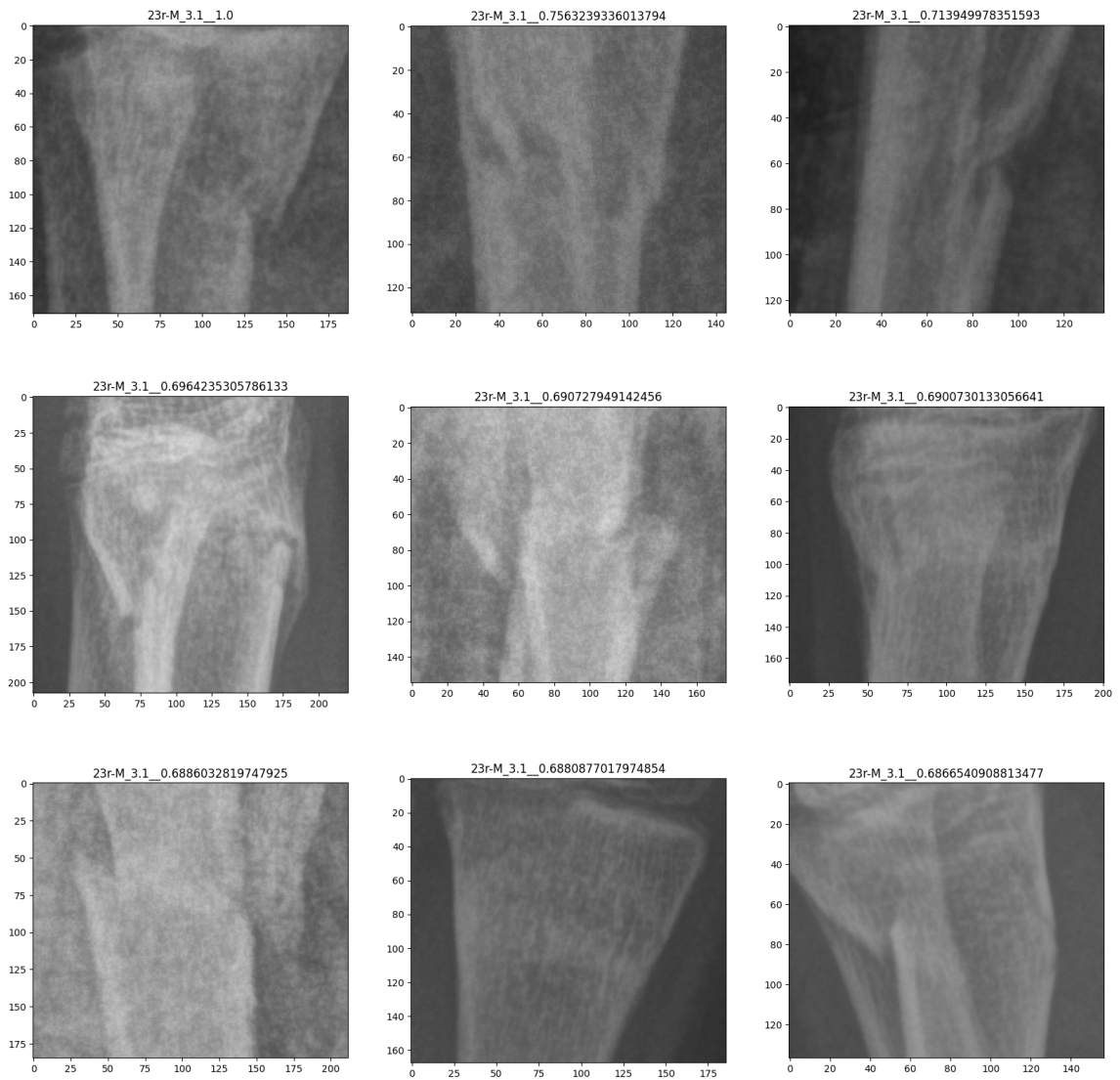
Slika 3.5 Primjer podudaranja slika s modelom trojke AttentionUnet 0.95 FC2\_4 4\_sloja\_1.

### Poglavlje 3. Rezultati



Slika 3.6 Primjer podudaranja slika s klasifikacijskim modelom AttentionUnet 0.95 FC2\_3\_3\_sloja\_2.

### Poglavlje 3. Rezultati



Slika 3.7 Primjer podudaranja slika s klasifikacijskim modelom UNet3+ 0.95 FC2\_3 3\_sloja\_2.

# Poglavlje 4

## Diskusija

Tijekom treniranja FE i FM stopa učenja od 0.00001 pokazala se najbolja u skoro svim modelima. Jedina iznimka je bila UNet mreža kod koje je manja stopa učenja dala bolje performanse. Za FE AttentionUNet i UNet3+ su se pokazali kao najbolji modeli.

Dok je UNet3+ imao bolje performanse tijekom treniranja, AttentionUNet je pokazao bolju sposobnost generalizacije zbog čega se smatra da ima bolju procjenu među klasne udaljenosti čineći ga boljim kandidatom za problem traženja sličnih rendgenskih slika.

Također, pokazalo se kako nije potreban velik broj podataka za uspješno pronalaženje sličnih slika već se optimalne performanse mogu postići i sa samo nekolicinom (100) slika.

Vrlo vjerojatan razlog sub optimalnih performansi je sub optimalan proces anotacije. Kao što je bilo spomenuto u poglavlju 2.1, frakture nisu anotirane već su anotirane slike na kojima se one nalaze, zbog čega se ne zna lokacija anotacija na slikama. Osim problema pripadnosti anotacija i fraktura, postoji i problem nejednako broja anotacija i fraktura na slikama. Ako je broj fraktura jednak broju klasa opisani algoritam za razvrstavanje anotacija po frakturama slike će funkcionirati. U protivnom, moguće je da dođe do pogreške u dodijeli anotacije.

Tri su slučaja u kojima algoritam griješi i svi se dešavaju radi broja fraktura i anotacija na lijevoj ili desnoj strani slike.

## Poglavlje 4. Diskusija

1. Kada slika sadrži više anotacija nego fraktura. Na primjer ako je slika anotirana s tri frakture dok su na njoj označene dvije frakture algoritam svrstavanja fraktura će spojiti dvije najviše anotacije s frakturama. Zbog toga anotacija, koja bi se trebala nalaziti pri vrhu kosti, može biti spojena s frakturom koja se nalazi na sredini kosti.
2. Kada je anotacija manje nego fraktura. Na primjer ako su na rendgenskoj slici dvije anotacije a tri frakture, algoritam pretpostavlja da su zadnje dvije frakture iste klase kao zadnja anotacija. Pošto klase anotacija nisu povezane za lokaciju na slici, teško je riješiti i jedan od dva navedena problema osim izbacivanjem slika iz baze podataka ili ručnim pregledavanjem rendgenskih slika.
3. Kada postoji samo jedna anotacija na slici za bilo koji broj fraktura. Pretpostavka algoritma je da su sve frakture na slici iste klase iako to ne mora biti slučaj. Pogotovo u situacijama s klasama bez oznake kosti  $r$  i  $u$  kao na primjer 23-M\_2.1. Te klase označavaju i radius i ulnu te bi trebale biti razdvojene na dvije zasebne klase kao što su razdvojene u slučajevima s više od jedne anotacije.

Treći problem je rješiv relativno laganom modifikacijom algoritma svrstavanja. Međutim, treći problem je primijećen tek nakon treniranja većine modela zbog čega je ostao ne riješen kako bi se zadržali isti uvjeti treniranja na svim modelima. Ipak, klasifikacijski modeli i modeli trojke su pokazali obećavajuće rezultate.

Modeli uspješno pronalaze slike visoke sličnosti s promatranom slikom te su performanse modela na nadklasama visoke unatoč suboptimalnim podacima. Postigao se značajan uvid u korištenje rekonstrukcijskog autoenkodera kao ekstraktora značajki i u sposobnosti raznovrsnih arhitektura modela sličnosti.

# Poglavlje 5

## Zaključak

U ovom diplomskom radu predlaže se nova metoda za pronalaženje sličnih medicinskih slika pomoću značajki. Metoda je testirana na pedijatrijskim podacima rendgenskih fraktura zgloba s devet klasa i tri natklase.

U svrhu anotiranja fraktura, razvijen je jednostavni sustav temeljen na pravilima. Pokazalo se da razvijeni sustav sadrži nekoliko mana koje se daju jednostavno ispraviti. Većina mana sustava proizlazi iz manjka informacija o poziciji anotacijama. Jedina pozicijska informacija koju anotacije sadrže je međusobna visina anotacija u odnosu na ruku. To jest, anotacije klase 'E' su iznad anotacija klase 'M' koje su iznad anotacija klase 'D'.

Razvijena je metoda za pronalaženje sličnih medicinskih slika pomoću značajki. Sastoji se od dvije duboke neuronske mreže. Prva duboka neuronska mreža je rekonstrukcijski autoenkoder korišten kao ekstraktor značajki. Korištena metrika za rekonstrukciju je Sørensen–Dice koeficijent. Isprobane su četiri vrste CNN autoenkodera. Svi isprobani CNN autoenkoderi uspješno rekonstruiraju slike sa Sørensen–Diceovim koeficijentom od 80 – 99.9% što je odraženo u velikoj sličnosti između originalne slike i rekonstruirane slike.

Drugi dio metode je duboka neuronska mreža sa svrhom grupiranja isto klasnih podataka i stvaranjem što veće međuklasne udaljenosti. Isprobane su dvije vrste modela: klasifikacijski model i model trojke. Klasifikacijski model je treniran s funkcijom greške unakrsna entropija, a model trojke je treniran s funkcijom greške poluograni-

## Poglavlje 5. Zaključak

čavajućom selekcijom trojki, za koju je razvijena nova metoda miješanja podataka. Kao procjenu udaljenosti podataka isprobane su metrike kosinusova sličnost i MSE. Točnost modela je korištena kao metrika performanse i ona predstavlja učestalost podudaranja klase promatrane slike s klasama najbližih slika.

Koristeći testne podatke kao promatrane slike, a podatke za treniranje kao bazu pretrage, modeli su postigli slične performanse od 38 – 40% točnosti na potklasama te 78 – 79% točnosti na natklasama. Dobivene performanse su slične performansama modela tijekom treniranja, zbog čega se može zaključiti da modeli posjeduju dobre sposobnosti generalizacije. Sve pronađene slike s oba tipa modela sadrže visoki postotak sličnosti s promatranom slikom neovisno o podudaranju klasa.

Pokazalo se kako je i mala količina podataka dovoljna za uspješno pronalaženje fraktura iste klase kao promatrana fraktura. Vizualnim pregledom pronađenih slika uočava se sličnost između slika te navedene metode uz pravilan skup podataka mogu polučiti izvanredne rezultate.

### 5.1 Budući rad

U budućem radu se planira popraviti sustav za anotiranje kako bi se model i njegove buduće iteracije mogle istrenirati i testirati na ispravno anotiranim podacima čiju će ispravnost dodatno validirati medicinski eksperti budući da je provedeno istraživanje iznjedrilo upravo oznake kao jedan od temeljnih problema.

Dodavanjem informacije o projekciji, strani slike i poziciji frakture mogla bi se povećati raznolikost postojećih podataka te time i pojednostaviti problematiku pronalaženja točnih fraktura. Osim dodavanja informacija iz anotacija, treba razmotriti i utjecaj obogaćivanja reprezentativnog vektora kombinacijom dvaju ili više različitih ekstraktora značajki.

Radi karakteristika mehanike pažnje, vrijedi razmotriti potencijalnu nadogradnju modela podudaranja značajki dodavanjem mehanike pažnje. Posljednje područje koje bi se potencijalno moglo poboljšati jest računanje sličnosti podataka. Postoje dodane metrike poput "metrike učenja" (*engl. Metric Learning*) [16] prema kojoj model sam uči računati sličnost podataka te se bolje prilagoditi naučenoj formuli.

# Bibliografija

- [1] E. Chang, K. Goh, G. Sychay, and G. Wu, "Cbsa: content-based soft annotation for multimodal image retrieval using bayes point machines," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 1, pp. 26–38, 2003.
- [2] M. Alkhawlani, M. Elmogy, and H. El Bakry, "Text-based, content-based, and semantic-based image retrievals: a survey," *Int. J. Comput. Inf. Technol*, vol. 4, no. 01, pp. 58–66, 2015.
- [3] A. Latif, A. Rasheed, U. Sajid, J. Ahmed, N. Ali, N. I. Ratyal, B. Zafar, S. H. Dar, M. Sajid, and T. Khalil, "Content-based image retrieval and feature extraction: a comprehensive review," *Mathematical Problems in Engineering*, vol. 2019, 2019.
- [4] I. M. Hameed, S. H. Abdulhussain, and B. M. Mahmmod, "Content-based image retrieval: A review of recent trends," *Cogent Engineering*, vol. 8, no. 1, p. 1927469, 2021.
- [5] W. Chen, Y. Liu, W. Wang, E. Bakker, T. Georgiou, P. Fieguth, L. Liu, and M. S. Lew, "Deep learning for instance retrieval: A survey," *arXiv preprint arXiv:2101.11282*, 2021.
- [6] P. Liu, J.-M. Guo, K. Chamnongthai, and H. Prasetyo, "Fusion of color histogram and lbp-based features for texture image retrieval and classification," *Information Sciences*, vol. 390, pp. 95–111, 2017.
- [7] D. Jiang and J. Kim, "Image retrieval method based on image feature fusion and discrete cosine transform," *Applied Sciences*, vol. 11, no. 12, p. 5701, 2021.
- [8] R. Ashraf, M. Ahmed, U. Ahmad, M. A. Habib, S. Jabbar, and K. Naseer, "Mdcbir-mf: multimedia data for content-based image retrieval by using multiple features," *Multimedia tools and applications*, vol. 79, no. 13, pp. 8553–8579, 2020.



## Bibliografija

- [9] A. Irtaza and M. A. Jaffar, "Categorical image retrieval through genetically optimized support vector machines (gosvm) and hybrid texture features," *Signal, Image and Video Processing*, vol. 9, no. 7, pp. 1503–1519, 2015.
- [10] C. Li, Y. Huang, and L. Zhu, "Color texture image retrieval based on gaussian copula models of gabor wavelets," *Pattern Recognition*, vol. 64, pp. 118–129, 2017.
- [11] R. Ashraf, M. Ahmed, S. Jabbar, S. Khalid, A. Ahmad, S. Din, and G. Jeon, "Content based image retrieval by using color descriptor and discrete wavelet transform," *Journal of medical systems*, vol. 42, no. 3, pp. 1–12, 2018.
- [12] W. Zhou, H. Li, J. Sun, and Q. Tian, "Collaborative index embedding for image retrieval," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 1154–1166, 2017.
- [13] N. Ali, K. B. Bajwa, R. Sablatnig, S. A. Chatzichristofis, Z. Iqbal, M. Rashid, and H. A. Habib, "A novel image retrieval based on visual words integration of sift and surf," *PloS one*, vol. 11, no. 6, p. e0157428, 2016.
- [14] M. Yousuf, Z. Mehmood, H. A. Habib, T. Mahmood, T. Saba, A. Rehman, and M. Rashid, "A novel technique based on visual words fusion analysis of sparse features for effective content-based image retrieval," *Mathematical Problems in Engineering*, vol. 2018, 2018.
- [15] Z. Mehmood, S. M. Anwar, N. Ali, H. A. Habib, and M. Rashid, "A novel image retrieval based on a combination of local and global histograms of visual words," *Mathematical Problems in Engineering*, vol. 2016, 2016.
- [16] N. Garcia and G. Vogiatzis, "Learning non-metric visual similarity for image retrieval," *Image and Vision Computing*, vol. 82, pp. 18–25, 2019.
- [17] A. Babenko and V. Lempitsky, "Aggregating local deep features for image retrieval," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1269–1277.
- [18] T. M. Lehmann, M. O. Güld, C. Thies, B. Fischer, K. Spitzer, D. Keysers, H. Ney, M. Kohlen, H. Schubert, and B. B. Wein, "Content-based image retrieval in medical applications," *Methods of information in medicine*, vol. 43, no. 04, pp. 354–361, 2004.
- [19] N. A. M. Zin, R. Yusof, S. A. Lashari, A. Mustapha, N. Senan, and R. Ibrahim, "Content-based image retrieval in medical domain: a review," in *journal of physics: conference series*, vol. 1019, no. 1. IOP Publishing, 2018, p. 012044.

## Bibliografija

- [20] H. Müller, N. Michoux, D. Bandon, and A. Geissbuhler, “A review of content-based image retrieval systems in medical applications—clinical benefits and future directions,” *International journal of medical informatics*, vol. 73, no. 1, pp. 1–23, 2004.
- [21] K. H. Hwang, H. Lee, and D. Choi, “Medical image retrieval: past and present,” *Healthcare informatics research*, vol. 18, no. 1, pp. 3–9, 2012.
- [22] A. Qayyum, S. M. Anwar, M. Awais, and M. Majid, “Medical image retrieval using deep convolutional neural network,” *Neurocomputing*, vol. 266, pp. 8–20, 2017.
- [23] A. E. Minarno, K. M. Ghufron, T. S. Sabrila, L. Husniah, and F. D. S. Sumadi, “Cnn based autoencoder application in breast cancer image retrieval,” in *2021 International Seminar on Intelligent Technology and Its Applications (ISITIA)*. IEEE, 2021, pp. 29–34.
- [24] Y. Cao, S. Steffey, J. He, D. Xiao, C. Tao, P. Chen, and H. Müller, “Medical image retrieval: a multimodal approach,” *Cancer informatics*, vol. 13, pp. CIN–S14 053, 2014.
- [25] J. Kalpathy-Cramer, W. Hersh *et al.*, “Automatic image modality based classification and annotation to improve medical image retrieval,” in *Medinfo*, 2007, pp. 1334–1338.
- [26] E. Nagy, M. Janisch, F. Hržić, E. Sorantin, and S. Tschauner, “A pediatric wrist trauma x-ray dataset (grazpedwri-dx) for machine learning,” *Scientific Data*, vol. 9, no. 1, pp. 1–10, 2022.
- [27] M. Farajzadeh-Zanjani, R. Razavi-Far, and M. Saif, “A critical study on the importance of feature extraction and selection for diagnosing bearing defects,” in *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2018, pp. 803–808.
- [28] G. N. Ramadevi, K. U. Rani, and D. Lavanya, “Importance of feature extraction for classification of breast cancer datasets—a study,” *International Journal of Scientific and Innovative Mathematical Research*, vol. 3, no. 2, pp. 763–368, 2015.
- [29] S. Khalid, T. Khalil, and S. Nasreen, “A survey of feature selection and feature extraction techniques in machine learning,” in *2014 science and information conference*. IEEE, 2014, pp. 372–378.

## Bibliografija

- [30] D. Bank, N. Koenigstein, and R. Giryes, “Autoencoders,” *arXiv preprint arXiv:2003.05991*, 2020.
- [31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.
- [32] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [33] M. Yin, P. Wang, C. Ni, and W. Hao, “Cloud and snow detection of remote sensing images based on improved unet3+,” 2022.
- [34] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu, “Unet 3+: A full-scale connected unet for medical image segmentation,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 1055–1059.
- [35] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, “Attention u-net: Learning where to look for the pancreas,” *arXiv preprint arXiv:1804.03999*, 2018.
- [36] “Towards Data Science convolutional autoencoders for image noise reduction,” <https://towardsdatascience.com/convolutional-autoencoders-for-image-noise-reduction-32fce9fc1763>, accessed: 2022-10-27.
- [37] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [38] L. Shahriyari, “Effect of normalization methods on the performance of supervised learning algorithms applied to htseq-fpkm-uniq data sets: 7sk rna expression as a predictor of survival in patients with colon adenocarcinoma,” *Briefings in Bioinformatics*, vol. 20, no. 3, pp. 985–994, 2019.
- [39] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [40] L. Prechelt, “Early stopping-but when?” in *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69.

## Bibliografija

- [41] K. Fukushima and S. Miyake, “Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition,” in *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.
- [42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [43] Y. LeCun *et al.*, “Lenet-5, convolutional neural networks,” *URL: <http://yann.lecun.com/exdb/lenet>*, vol. 20, no. 5, p. 14, 2015.
- [44] prof. dr. sc. Ivan Štajduhar, “Predavanje kolegija strojno učenje,” Merlin sustav za e-učenje, 2020./2021.
- [45] G. Sumbul, M. Ravanbakhsh, and B. Demir, “Informative and representative triplet selection for multi-label remote sensing image retrieval,” *IEEE Transactions on Geoscience and Remote Sensing*, 2022.
- [46] B. Yu, T. Liu, M. Gong, C. Ding, and D. Tao, “Correcting the triplet selection bias for triplet loss,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 71–87.
- [47] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

# Pojmovnik

**AVG** Average Pooling Layer. 12, 17, 18, 28

**BCE** Binary Cross-Entropy. 15, 20, 26

**BoVW** Bag of Visual Words. 2

**CBIR** Content Based Image Retrieval. 1–3

**CBMIR** Content Based Medical Image Retrieval. 3

**CIR** Category level Image Retrieval. 2, 9, 16

**CNN** Convolutional Neural Network. 2, 3, 12, 17, 18, 26–30, 32, 34, 36, 37, 46, 55, 56

**DCNN** Deep Convolutional Neural Network. 3

**DNN** Deep Artificial Neural Network. 2, 16

**FC** Fully Connected Artificial Neural Network. 17, 18, 28, 31, 32

**FE** Feature Extractor. vi, 1, 2, 10, 14–16, 19, 20, 23, 28, 31–37, 44, 59

**FM** Feature Matching. 1, 2, 10, 16, 17, 19, 23, 32–34, 37, 38, 44

**FN** False Negative. 23–25

**FP** False Positive. 23, 24

**FS** Feedforward scheme. 1

**IIR** Instance level Image Retrieval. 2

**IR** Image Retrieval. 1

**MSE** Mean Squard Error. 24, 25, 34, 38–40, 47, 59, 60

**PACS** Picture Archiving and Communication Systems. 3

**PCA** Principal Component Analysis. 16, 17, 19, 31, 32, 59

**ReLU** Rectified Linear Activation Unit. 17, 19

**SIFT** Scale-Invariant Feature Transform. 2

**TBIR** Text Based Image Retrieval. 1

**TP** True Positive. 23–25

# Sažetak

Ovaj diplomski rad predlaže novu metodu za pronalaženje sličnih medicinskih slika pomoću značajki. Napravljen je pregled radova i metoda za pronalaženje sličnih slika pomoću značajki. Predstavljena je korištena medicinska pedijatrijska baza podataka fraktura za koju je razvijen jednostavni sustav za anotiranje fraktura. Osmišljena je metoda koja se sastoji od ekstraktora značajki i modela sličnosti. Isprobana su četiri CNN autoenkodera za ekstrakciju značajki. Model sličnosti pretvara ekstrahirane značajke u reprezentativni vektor koji je uspoređivan s drugim vektorima pomoću matematičke metode. Dvije metode treniranja modela sličnosti su istražene i objašnjene. Naposljetku, predstavljeni su rezultati, korištene metrike i svi problemi na koje se naišlo tijekom razvijanja metode.

***Ključne riječi*** — **Strojno učenje, Pronalaženje sličnih medicinskih, CNN autoenkoder slika pomoću značajki, Medicinske pedijatrijske slike fraktura**

## Abstract

This thesis proposes a new method for finding similar medical images using features. A review of papers and methods for finding similar images using features is presented. A medical pediatric fracture database is used for which a simple fracture annotation system was developed and presented. A method consisting of a feature extractor and a similarity model was devised. Four CNN autoencoders were tested for feature extraction. The similarity model transforms the extracted features into a representative vector that is compared with other vectors using a mathematical method. Two methods for training similarity models are explored and explained. Finally, the results, the metrics used and all the problems encountered during the development of the method are presented.

***Keywords*** — Machine learning, Content based medical image retrieval, CNN autoencoder, Images of medical pediatric fractures



# Popis slika

2.1	Primjeri slika s različitim vrijednostima projekcije i strane. . . . .	5
2.2	Raspodjela rendgenskih slika fraktura zgloba ruke po projekciji i oznaci strane. . . . .	6
2.3	Prikaz kostiju radius i ulna na lijevoj i desnoj ruci. . . . .	8
2.4	Primjeri razvrstavanja klasa fraktura po označenim frakturama. . . .	9
2.5	Koncept za detekciju sličnih fraktura. . . . .	11
2.6	Korišteni enkoder dekodeer modeli. . . . .	13
2.7	Primjer konvolucije, preuzeto s [44]. . . . .	18
2.8	Primjer funkcije gubitka trojki, preuzeto sa [45]. Lijevo su prikazana tri para pozitivnih i negativnih primjeraka prije učenja pomoću funkcije gubitka trojki. $P_i$ i $N_i$ predstavljaju pozitivni i negativni par primjeraka u trojci dok je $x_a$ sidro trojke. Desno je prikazana ciljana situacija udaljenosti među primjercima nakon učenja s funkcijom gubitka trojki. . . . .	21
3.1	Primjeri rekonstrukcije slika s najboljim modelima iz tablica 3.1 i 3.2.	29
3.2	Primjeri rekonstrukcije slika s najboljim modelima iz tablica 3.1 i 3.2.	30
3.3	Iznimke jednakosti najbolje rekonstrukcije i najboljeg Sørensen–Dice koeficijenta kod AttentionUnet modela. . . . .	31
3.4	Matrice zabune najboljih klasifikacijskih modela na validacijskim i testnim podacima. . . . .	35

3.5	Primjer podudaranja slika s modelom trojke AttentionUnet 0.95 FC2_4 4_sloja_1. . . . .	41
3.6	Primjer podudaranja slika s klasifikacijskim modelom AttentionUnet 0.95 FC2_3 3_sloja_2. . . . .	42
3.7	Primjer podudaranja slika s klasifikacijskim modelom UNet3+ 0.95 FC2_3 3_sloja_2. . . . .	43
A.1	CNN Autoenkoder [36]. . . . .	61
A.2	UNet [32]. . . . .	62
A.3	UNet3+ [34]. . . . .	62
A.4	AttentionUNet [35]. . . . .	63
B.1	Primjer podudaranja slika s modelom trojke AttentionUnet 0.95 FC2_4 4sloja_1. . . . .	65
B.2	Primjer podudaranja slika s klasifikacijskim modelom AttentionUnet 0.95 FC2_3 3sloja_2. . . . .	66
B.3	Primjer podudaranja slika s klasifikacijskim modelom UNet3+ 0.95 FC2_3 3sloja_2. . . . .	67
B.4	Primjer podudaranja slika s modelom trojke AttentionUnet 0.95 FC2_4 4sloja_1. . . . .	68
B.5	Primjer podudaranja slika s klasifikacijskim modelom AttentionUnet 0.95 FC2_3 3sloja_2. . . . .	69
B.6	Primjer podudaranja slika s klasifikacijskim modelom UNet3+ 0.95 FC2_3 3sloja_2. . . . .	70

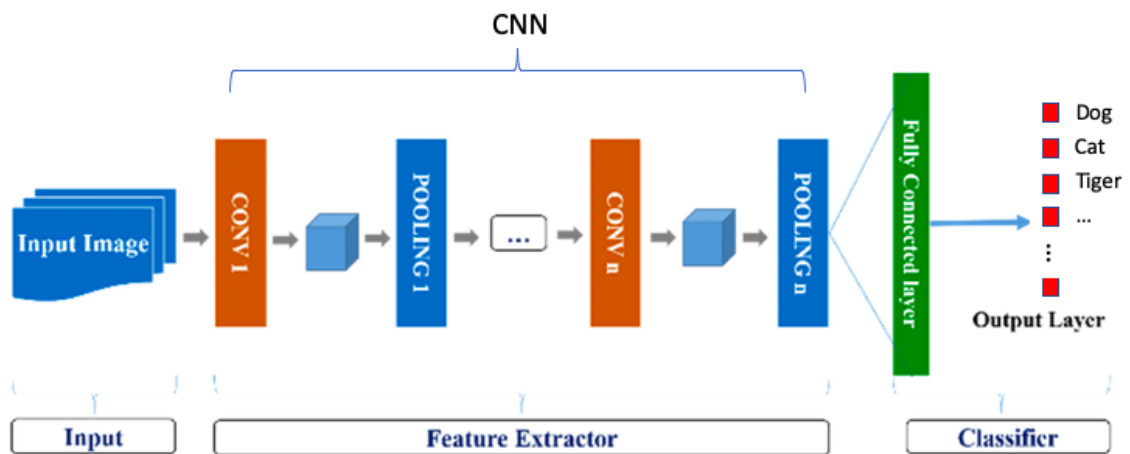
# Popis tablica

2.1	Prikaz preostalih klasa fraktura u bazi podataka nakon pročišćavanja.	7
2.2	Prikaz preostalih natklasa fraktura u bazi podataka nakon pročišćavanja. . . . .	7
3.1	Performanse rekonstrukcijskih modela. . . . .	27
3.2	Performanse pretreniranih rekonstrukcijskih modela. . . . .	28
3.3	Oznake gradivnih blokova arhitekture modela sličnosti. . . . .	28
3.4	Isprobane kombinacije FE i PCA. . . . .	32
3.5	Isprobane arhitekture modela sličnosti i njihove oznake. . . . .	32
3.6	Brojevi neurona korišteni u skrivenim slojevima i oznaka kombinacije.	33
3.7	Najbolje performanse klasifikacijskih modela s 3 skrivena sloja. . . .	34
3.8	Najbolje performanse AttentionUNet klasifikacijskih modela s tri, četiri i pet skrivenih slojeva. . . . .	34
3.9	Performanse najboljih klasifikacijskih modela nad testnim podacima.	36
3.10	Performanse modela trojke s tri skrivena sloja. . . . .	37
3.11	Performanse modela trojke AttentionUNet s tri, četiri ili pet skrivenih slojeva. . . . .	37
3.12	Točnost najboljih modela podudaranja slika koristeći metrike kosinusovu sličnost i MSE. . . . .	39
3.13	Točnost klasifikacijskog modela UNet3+ 0.95 FC2_3 3_sloja_2 u podudaranju natklasa slika koristeći metrike kosinusovu sličnost i MSE.	39

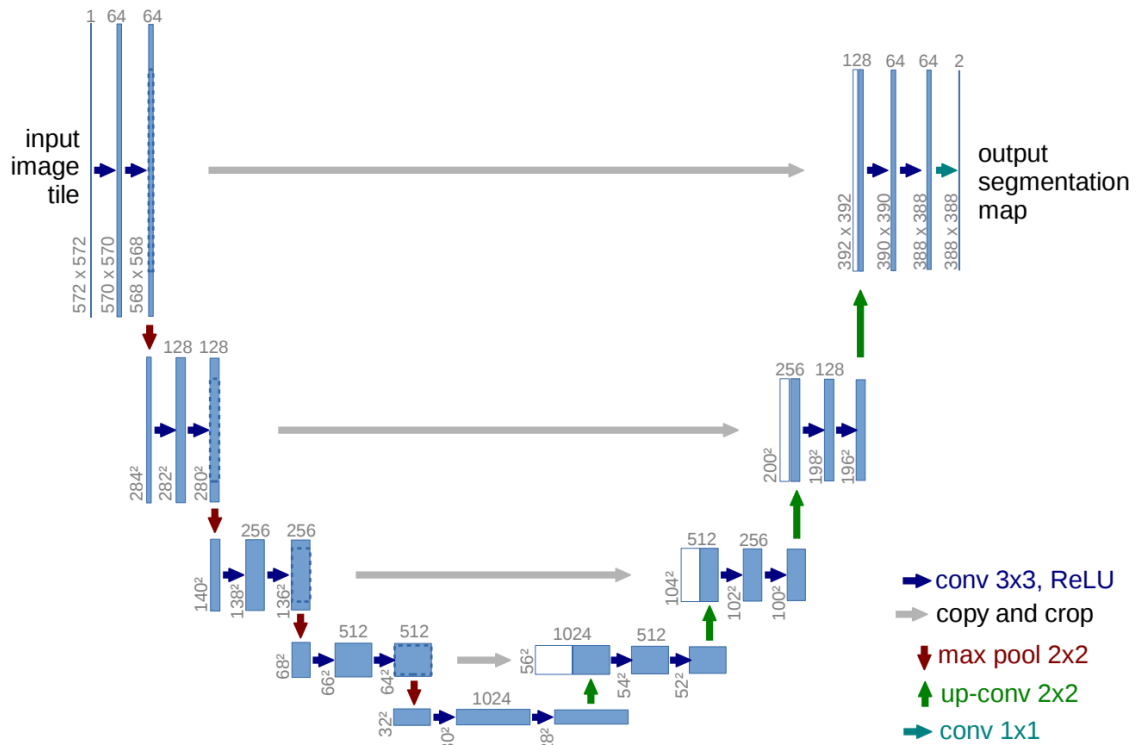
3.14	Točnost klasifikacijskog modela AttentionUnet 0.95 FC2_3 3_sloja_2 u podudaranju natklasa slika koristeći metrike kosinusovu sličnost i MSE. . . . .	39
3.15	Točnost modela trojke AttentionUnet 0.95 FC2_4 4_sloja_1 u podudaranju natklasa slika koristeći metrike kosinusovu sličnost i MSE. . . . .	40

# Dodatak A

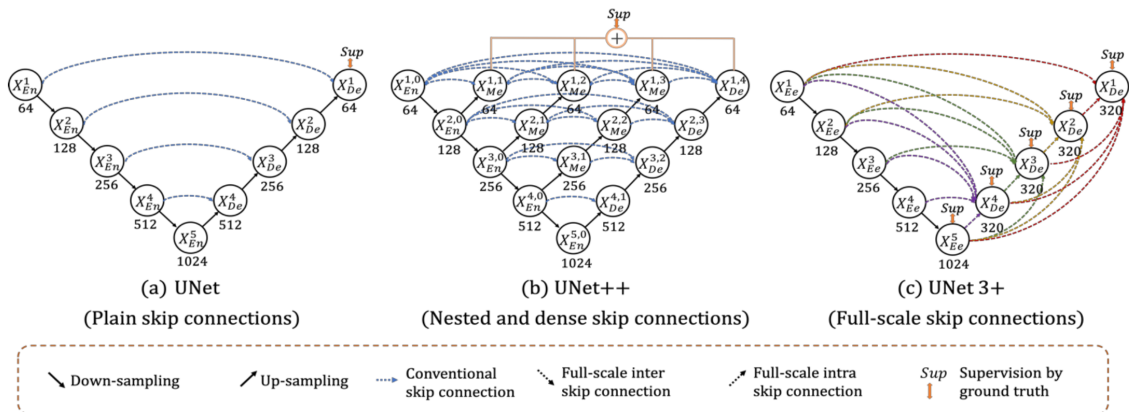
## Enkoder dekoder arhitecture



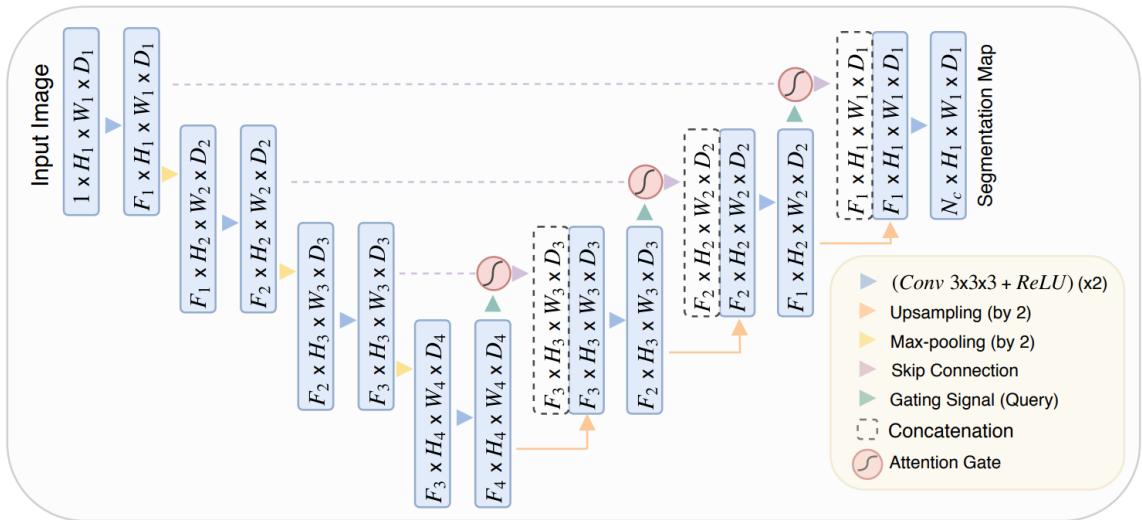
Slika A.1 CNN Autoenkoder [36].



Slika A.2 UNet [32].



Slika A.3 UNet3+ [34].

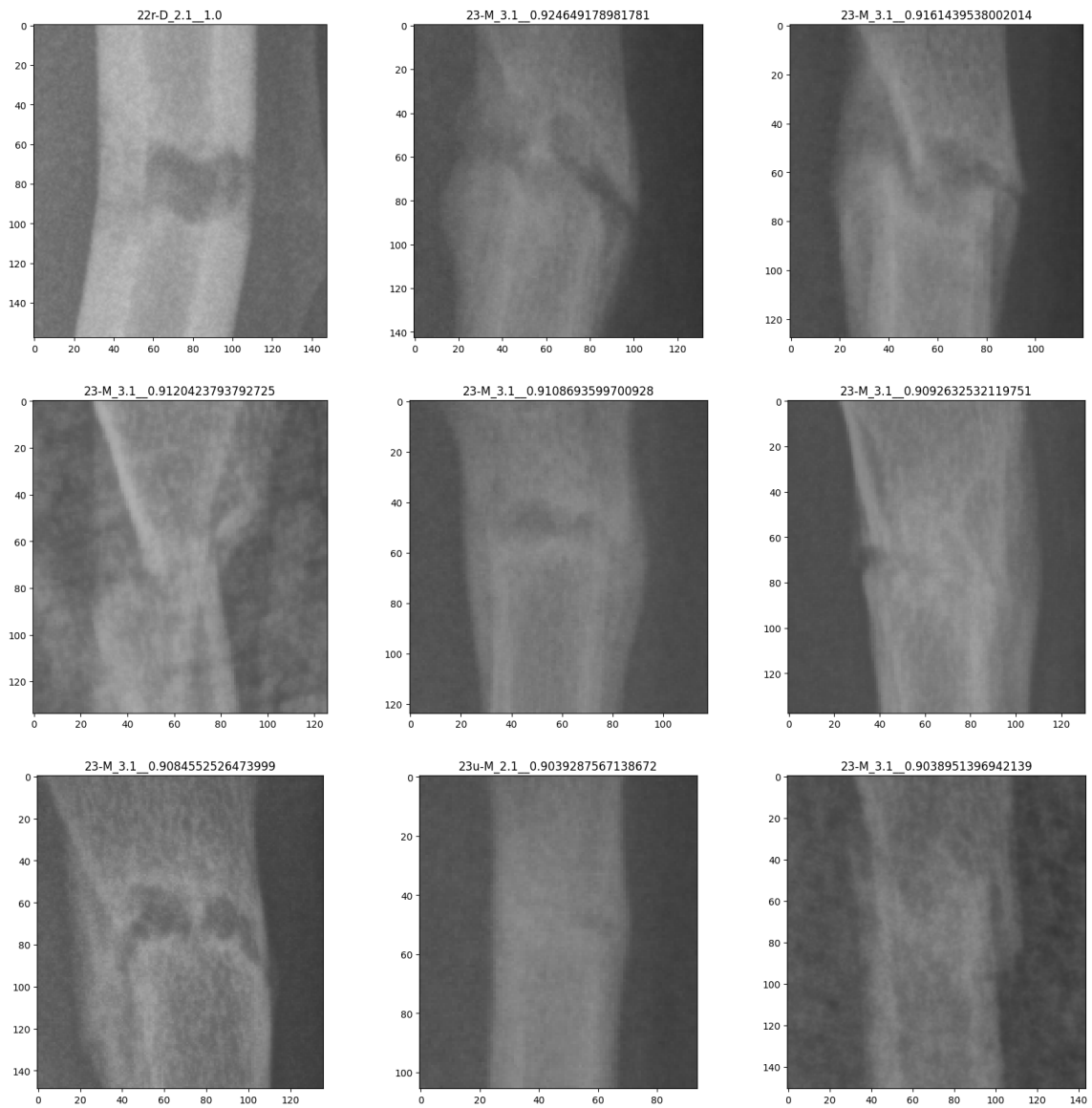


Slika A.4 AttentionUNet [35].

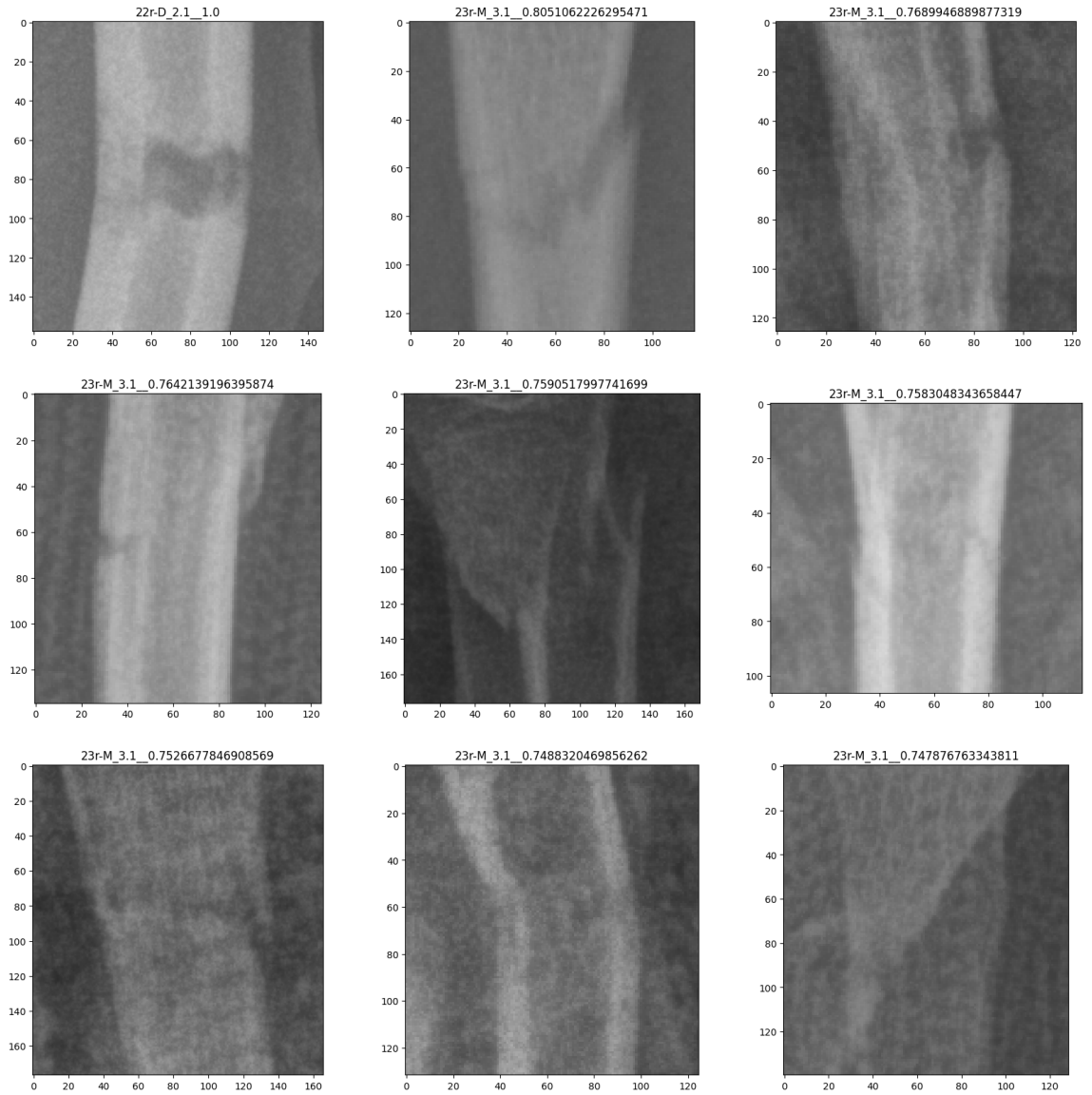
## Dodatak B

### Dodatni primjeri podudaranja slika

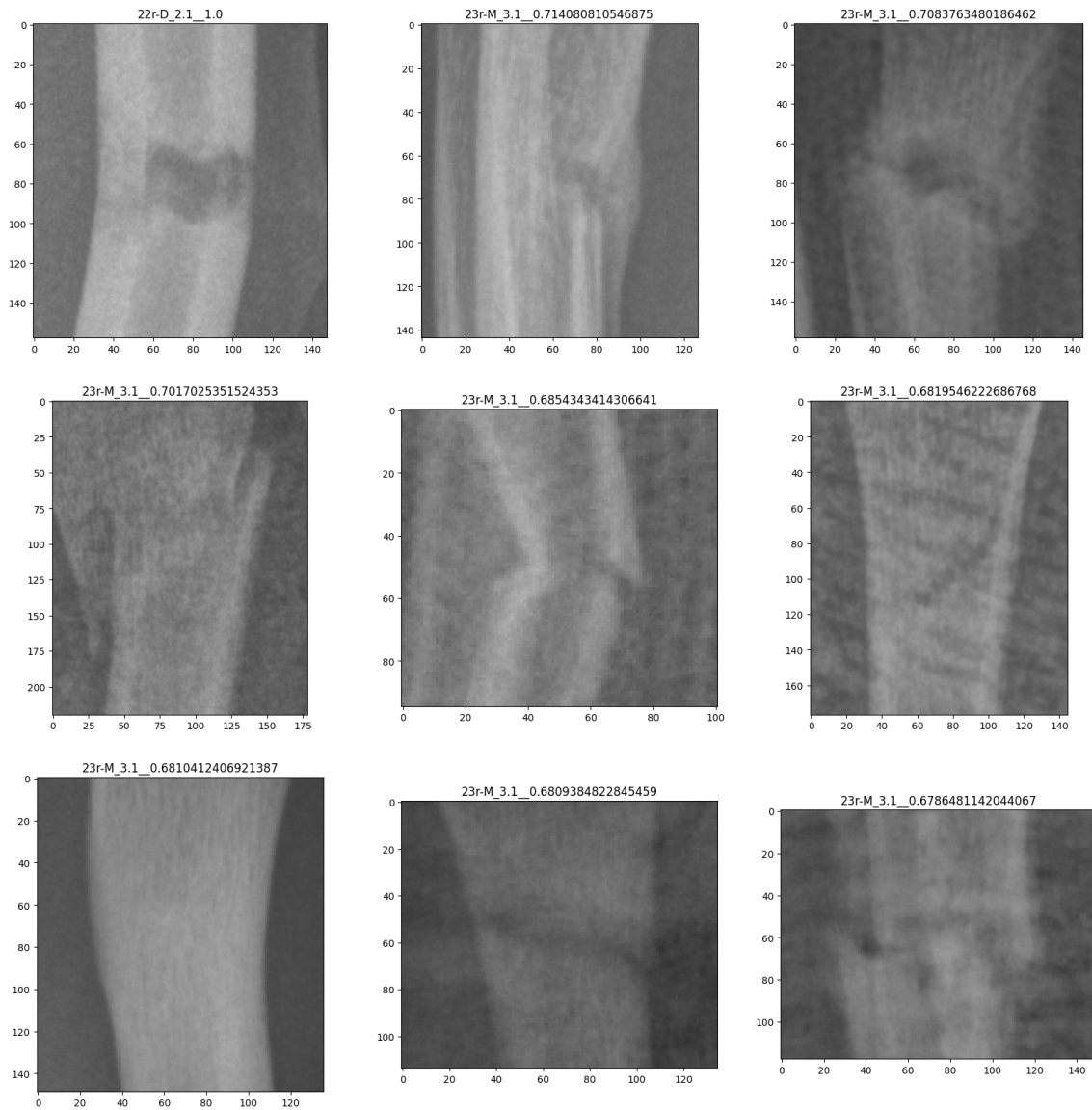




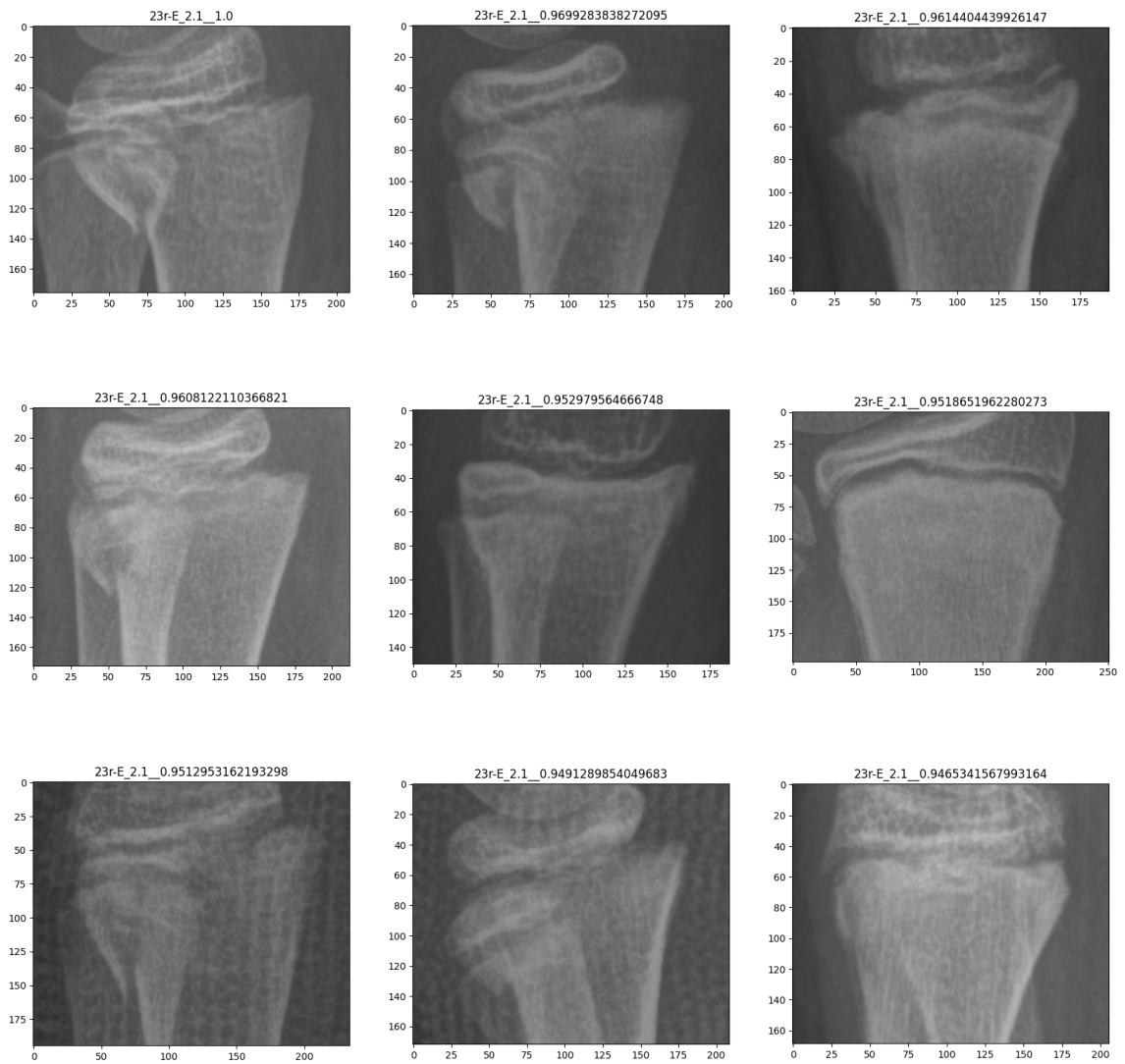
Slika B.1 Primjer podudaranja slika s modelom trojke AttentionUnet 0.95 FC2\_4 4sloja\_1.



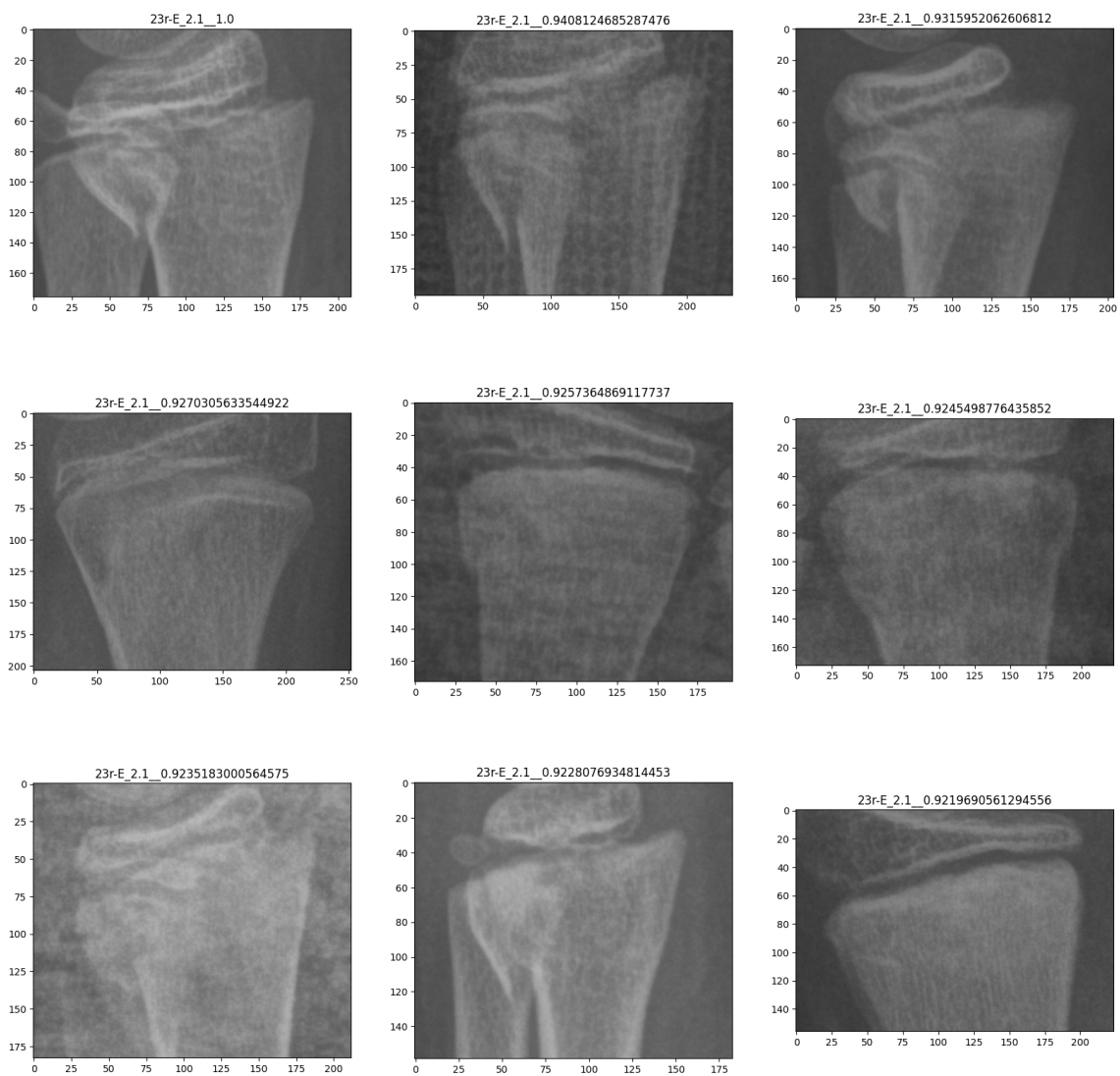
Slika B.2 Primjer podudaranja slika s klasifikacijskim modelom AttentionUnet 0.95 FC2\_3 3sloja\_2.



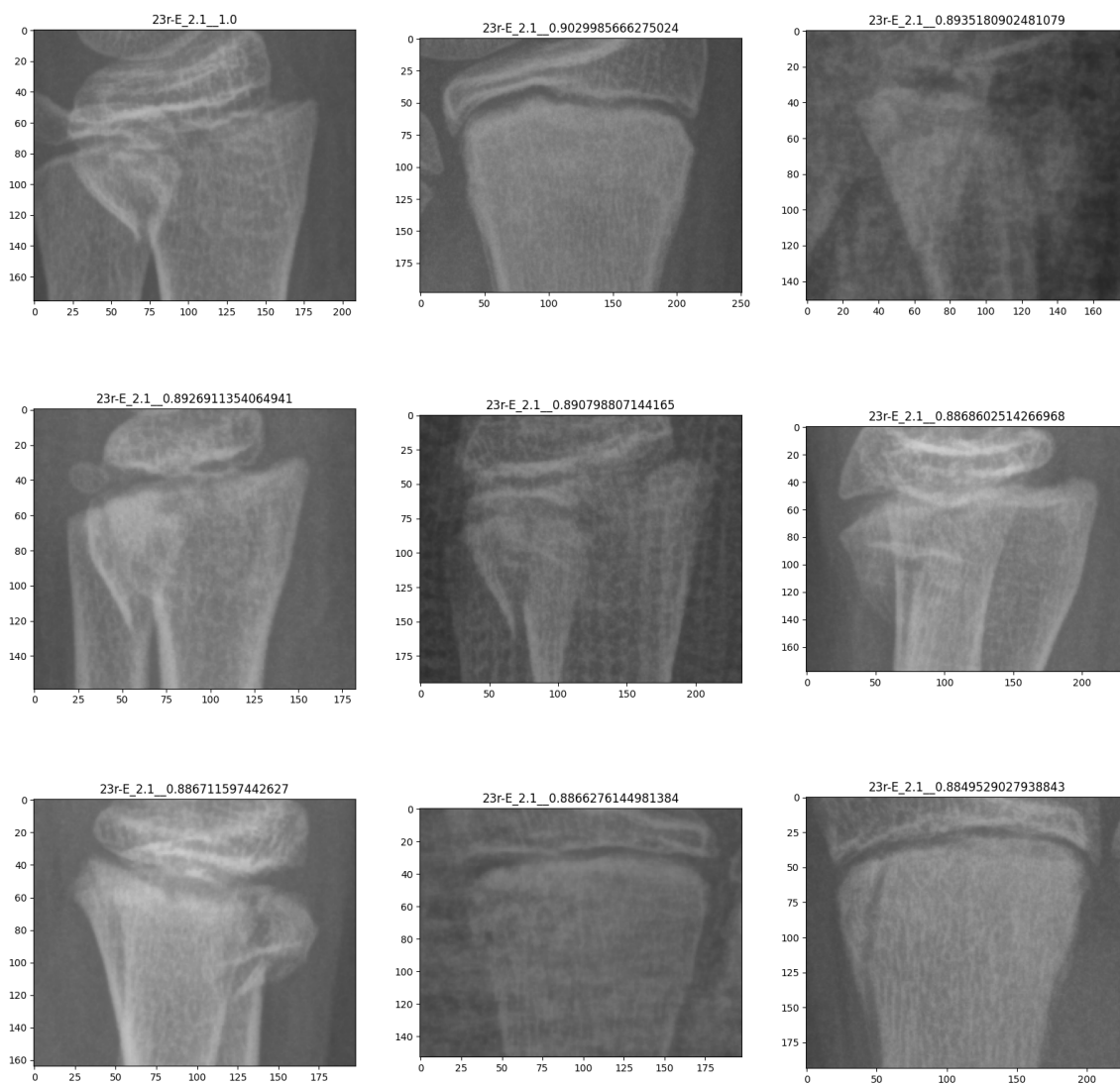
Slika B.3 Primjer podudaranja slika s klasifikacijskim modelom UNet3+ 0.95 FC2\_3 3sloja\_2.



Slika B.4 Primjer podudaranja slika s modelom trojke AttentionUnet 0.95 FC2\_4 4sloja\_1.



Slika B.5 Primjer podudaranja slika s klasifikacijskim modelom AttentionUnet 0.95 FC2\_3 3sloja\_2.



Slika B.6 Primjer podudaranja slika s klasifikacijskim modelom UNet3+ 0.95 FC2\_3 3sloja\_2.