

# Pretraživanje slika PACS sustava prema njihovom sadržaju

---

Sečen, Karlo

Undergraduate thesis / Završni 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://urn.nsk.hr/urn:nbn:hr:190:389138>

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

*Download date / Datum preuzimanja:* **2024-05-17**



*Repository / Repozitorij:*

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



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

Završni rad

**Pretraživanje slika PACS sustava prema  
njihovom sadržaju**

Rijeka, srpanj 2022.

Karlo Sečen  
0069082354

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

Završni rad

**Pretraživanje slika PACS sustava prema  
njihovom sadržaju**

Mentor: prof. dr. sc. Ivan Štajduhar

Rijeka, srpanj 2022.

Karlo Sečen  
0069082354

Umjesto ove stranice umetnuti zadatak  
za završni ili diplomski rad



## Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, srpanj 2022.

-----  
Karlo Sečen

# Zahvala

Zahvalio bih mentoru prof. dr. sc. Ivanu Štajduharu na svim savjetima, uputama i raspravama koje su mi pomogle u izradi ovog završnog rada.

Također, zahvalio bih svojoj obitelji i prijateljima na podršci kroz cijelo studentsko obrazovanje i Bogu, koji me čuva i vodi kroz cijeli život.

# Sadržaj

|  |             |
|--|-------------|
| <b>Popis slika</b>   | <b>viii</b> |
| <b>Popis tablica</b>   | <b>x</b>    |
| <b>1 Uvod</b>  | <b>1</b>    |
| <b>2 PACS sustav</b>   | <b>3</b>    |
| 2.1 DICOM standard . . . . .   | 4           |
| 2.1.1 Medicinska slika . . . . .                                       | 4           |
| 2.1.2 Metaoznake medicinske slike . . . . .                            | 5           |
| <b>3 Podatci</b>   | <b>9</b>    |
| 3.1 Podrijetlo podataka . . . . .                                      | 9           |
| 3.2 Način pohrane podataka . . . . .                                   | 10          |
| 3.3 Oprema i alati korišteni u radu . . . . .                          | 11          |
| 3.3.1 Računalna oprema . . . . .                                       | 11          |
| 3.3.2 Programski jezik i paketi . . . . .                              | 11          |
| <b>4 Pretraživanje slika PACS sustava prema njihovom sadržaju</b>      | <b>13</b>   |
| 4.1 Tehnike pretraživanja slika . . . . .                              | 13          |
| 4.2 Problem pretraživanja voluminoznih podataka PACS sustava . . . . . | 15          |

## *Sadržaj*

|          |   |           |
|----------|---|-----------|
| 4.3      | Prototip rješenja . . . . .             | 16        |
| 4.3.1    | Autokoder . . . . .                     | 17        |
| 4.3.2    | Algoritam k-najbližih susjeda . . . . . | 21        |
| 4.3.3    | Tražilica . . . . .                     | 23        |
| <b>5</b> | <b>Rezultati</b>                        | <b>27</b> |
| 5.1      | Rezultati prototipa rješenja . . . . .  | 27        |
| 5.2      | Vremenska analiza . . . . .             | 39        |
| <b>6</b> | <b>Zaključak</b>                        | <b>41</b> |
|          | <b>Bibliografija</b>                    | <b>42</b> |
|          | <b>Pojmovnik</b>                        | <b>44</b> |
|          | <b>Sažetak</b>                          | <b>45</b> |
| <b>A</b> | <b>Naslov priloga</b>                   | <b>46</b> |

# Popis slika

|      |  |    |
|------|--|----|
| 2.1  | <i>Slika dobivena pregledom rendgen uređaja (CR)</i> . . . . . | 5  |
| 2.2  | <i>Prikaz DICOM datoteke</i> . . . . .                         | 6  |
| 2.3  | <i>Prikaz strukture podatkovne jedinice</i> . . . . .          | 7  |
| 4.1  | <i>Struktura mreže autokodera</i> . . . . .                    | 18 |
| 4.2  | <i>Primjer autokodera</i> . . . . .                            | 19 |
| 4.3  | <i>Grafičko sučelje CBIR.py datoteke</i> . . . . .             | 24 |
| 4.4  | <i>Prozor "Show results (mosaic)"</i> . . . . .                | 26 |
| 5.1  | <i>Primjer 1 - medicinska slika</i> . . . . .                  | 28 |
| 5.2  | <i>Primjer 1 - rezultati</i> . . . . .                         | 28 |
| 5.3  | <i>Primjer 2 - medicinska slika</i> . . . . .                  | 29 |
| 5.4  | <i>Primjer 2 - rezultati</i> . . . . .                         | 29 |
| 5.5  | <i>Primjer 3 - medicinska slika</i> . . . . .                  | 30 |
| 5.6  | <i>Primjer 3 - rezultati</i> . . . . .                         | 30 |
| 5.7  | <i>Primjer 4 - medicinska slika</i> . . . . .                  | 31 |
| 5.8  | <i>Primjer 4 - rezultati</i> . . . . .                         | 31 |
| 5.9  | <i>Primjer 5 - medicinska slika</i> . . . . .                  | 32 |
| 5.10 | <i>Primjer 5 - rezultati</i> . . . . .                         | 32 |
| 5.11 | <i>Primjer 6 - medicinska slika</i> . . . . .                  | 33 |

## Popis slika

|      |  |    |
|------|--|----|
| 5.12 | <i>Primjer 6 - rezultati</i>                             | 33 |
| 5.13 | <i>Primjer 7 - medicinska slika</i>                      | 34 |
| 5.14 | <i>Primjer 7 - rezultati</i>                             | 34 |
| 5.15 | <i>Primjer 8 - medicinska slika</i>                      | 35 |
| 5.16 | <i>Primjer 8 - rezultati</i>                             | 35 |
| 5.17 | <i>Primjer 9 - medicinska slika</i>                      | 36 |
| 5.18 | <i>Primjer 9 - rezultati - Minkowskijeva udaljenost</i>  | 36 |
| 5.19 | <i>Primjer 9 - rezultati - kosinusna udaljenost</i>      | 37 |
| 5.20 | <i>Primjer 10 - medicinska slika</i>                     | 38 |
| 5.21 | <i>Primjer 10 - rezultati - Minkowskijeva udaljenost</i> | 38 |
| 5.22 | <i>Primjer 10 - rezultati - kosinusna udaljenost</i>     | 39 |

# Popis tablica

|     |   |    |
|-----|---|----|
| 3.1 | Prikaz detalja korištenih skupova podataka . . . . .      | 11 |
| 5.1 | Prikaz vremenskog trajanja operacija aplikacije . . . . . | 40 |

# Poglavlje 1

## Uvod

Pretraživanje slika dobivenih medicinskim pretragama može biti vrlo korisno u radu medicinskog osoblja kod dijagnostike i liječenja bolesti pacijenata te u radu znanstvenika iz područja umjetne inteligencije koji razvijaju računalne metode potrebne za prikupljanje, analizu i obradu podataka u medicini. Picture archiving and communication system (PACS)(engl.) je standardizirani sustav arhiviranja i pristupanja slikama dobivenih medicinskim pretragama koji omogućuje ekonomično skladištenje i praktičan pristup navedenim slikama [1]. Navedene slike u PACS sustavu spremaju se u Digital Imaging and Communications in Medicine (DICOM) formatu, kojeg čine skup vrijednosti svih piksela slike (ono što nazivamo slikom) i metaoznake vezane za tu sliku pretrage. Metaoznake čine sve informacije vezane za taj pregled, većina njih je automatski generirano od strane uređaja koji je obavio pregled, dok se ostatak ručno unosi od strane medicinskog osoblja.

No, pretraživanje slika u PACS sustavu nije uvijek jednostavno. Jedan od problema može biti veličina skupova podataka koji se skladište u PACS sustavima kliničkih centara, neki sustavi imaju nekoliko terabajta podataka generiranih medicinskim pretragama. Zatim, treba uzeti u obzir mogućnost krivog unošenja podataka u PACS sustav, ponajviše zbog ručnog unosa metaoznaka koje vrši medicinsko osoblje te koje može biti nestandardizirano ili netočno zbog faktora ljudske pogreške, što može uzrokovati nemogućnost automatske, računalne obrade unutar nekog većeg skupa podataka [2]. Zbog toga, vrlo je važno razmisliti o pristupu i načinu pretraživanja navedenih slika.



## *Poglavlje 1. Uvod*

Ovaj rad će opisati nekoliko tehnika pretraživanja slika te će koristiti pretraživanje slika po sadržaju (engl. content-based image retrieval) (CBIR) kao tehniku kojom će razviti prototip rješenja pretraživanja slika PACS sustava. Navedeno će pomoći u postizanju ciljeva projekta HRZZ IP-2020-02-3770 "Strojno učenje za prijenos znanja u medicinskoj radiologiji", a koji je izvor podataka i motivacije ovog završnog rada.

## Poglavlje 2

### PACS sustav

PACS (engl. picture archiving and communication system) je sustav za arhiviranje slika i komunikaciju koji omogućuje ekonomično pohranjivanje i praktičan pristup slikama iz više modaliteta snimanja za potrebe medicinskih pregleda. Elektroničke slike i izvješća prenose se digitalno putem PACS-a, čime se eliminira potreba za ručnim arhiviranjem, dohvaćanjem ili transportom omota filma, mapa koje se koriste za pohranjivanje i zaštitu rendgenskog filma. Univerzalni format za pohranu i prijenos slika u PACS sustavu je DICOM (engl. Digital Imaging and Communications in Medicine). Podaci koji nisu slikovni, kao što su skenirani dokumenti, mogu se uključiti korištenjem standardnih formata potrošačke industrije kao što je Portable Document Format (PDF), nakon što su uneseni (enkapsulirani) u DICOM format datoteke. PACS se sastoji od četiri glavne komponente: modaliteti snimanja kao što su rendgenski film (engl. X-ray plain film) (PF), računalna tomografija (engl. computed tomography) (CT), magnetska rezonancija (engl. magnetic resonance imaging) (MRI) i ostali, sigurna mreža za prijenos informacija o pacijentu, radna stanica za tumačenje i pregled slike i arhiva za pohranjivanje i dohvaćanje slika i izvješća. U kombinaciji s dostupnom i novom web tehnologijom, PACS ima mogućnost pravovremenog i učinkovitog pristupa slikama, interpretacijama i povezanim podacima. PACS smanjuje fizičke i vremenske barijere povezane s pronalaženjem, distribucijom i prikazom slike temeljene na tradicionalnom filmu.[1]

## 2.1 DICOM standard

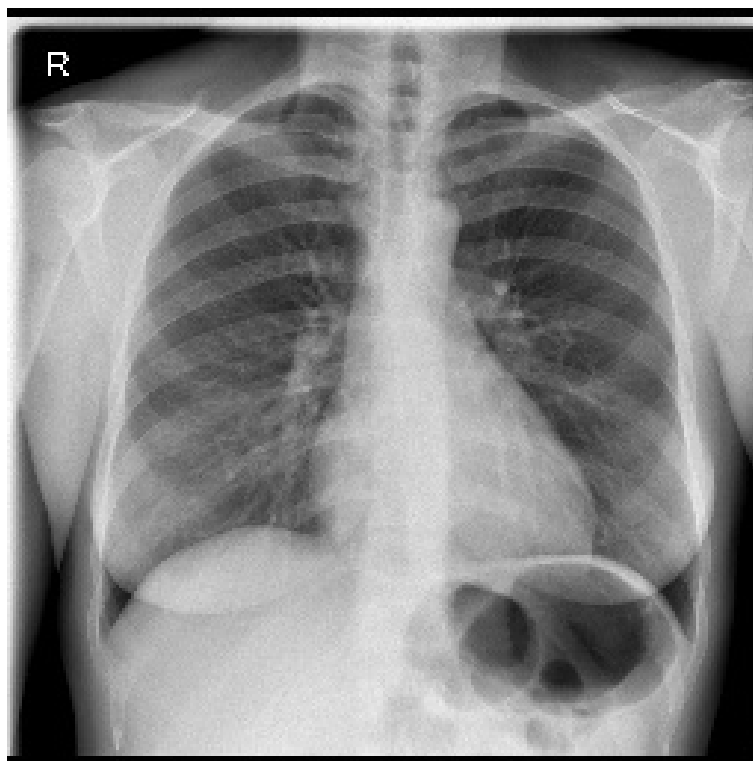
DICOM (engl. Digital Imaging and Communications in Medicine) međunarodni je standard za komunikaciju i upravljanje medicinskim slikama i povezanim podacima. Također, omogućava integraciju medicinskih uređaja za snimanje sa uređajima kao što su skeneri, poslužitelji, radne stanice, pisači, mrežni hardver i sustavi za arhiviranje slika i komunikaciju (PACS) koji mogu biti proizvedeni od različitih proizvođača. Bolnice su ga naširoko usvojile i sve se više implementira u manje medicinske ustanove kao što su zubarske i liječničke ordinacije. Standard su razvili American College of Radiology (ACR) i National Electrical Manufacturers Association (NEMA), najveće trgovačko udruženje proizvođača električne opreme u Sjedinjenim Državama, koje je danas i vlasnik autorskih prava na standard.[3][4][5]

DICOM datoteka sastoji se od velikog broja metaoznaka, neke čak i više od 4000 metaoznaka. Možemo ih podijeliti na dvije glavne grupe: metaoznake koje sadrže **medicinsku sliku** i ostale **metaoznake medicinske slike**.

### 2.1.1 Medicinska slika

Medicinska slika je reprezentativni prikaz rezultata medicinske pretrage, ono što nazivamo "slikom". Pohranjena je unutar metaoznake PixelData, pod adresom (7FE0,0010). Zapisana je kao skup vrijednosti svakog piksela koji čine samu sliku.

Redoslijed piksela kodiranih za svaku ravninu slike je slijeva nadesno, odozgo prema dolje, npr. prvi je kodiran gornji lijevi piksel (označen s 1,1), a zatim ostatak reda 1, a zatim prvi piksel reda 2 (označeno s 2,1), zatim ostatak reda 2 i tako dalje.



Slika 2.1 *Slika dobivena pregledom rendgen uređaja (CR)*

Sliku možemo prikazati na računalu ako je otvorimo u nekom specijaliziranom programu za pregled DICOM datoteka, poput onih koje imaju u kliničkim centrima ili u besplatnim "DICOM Viewer" programima koje možemo pronaći i na internetu. Također, moguće je sliku prikazati ako sami obradimo podatke zapisane u metaoznaci PixelData. Potrebno je izvući vrijednosti svih piksela iz točno definirane adrese kao matricu intenziteta piksela (engl. pixel value matrix). Ovisno o knjižnici ili implementaciji koju koristimo za prikazivanje slika, potrebno je i skaliranje matrice intenziteta piksela kako bi ispravno prikazali sve boje i nijanse unutar slike.

### 2.1.2 Metaoznake medicinske slike

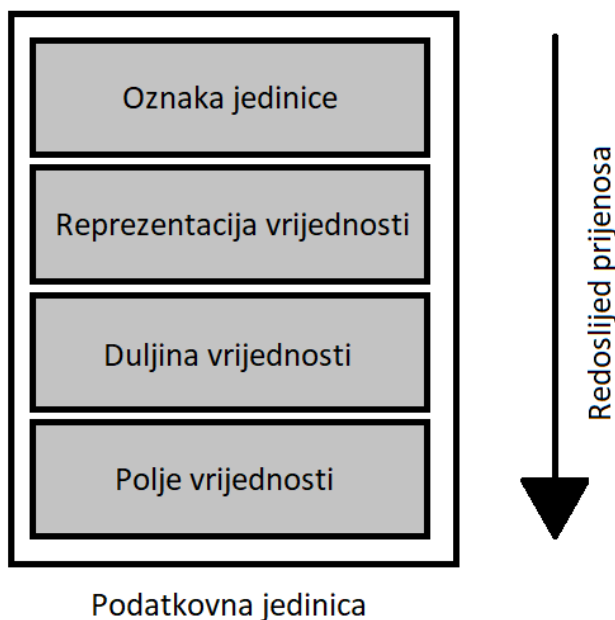
Uz metaoznaku PixelData zaduženu za prikaz slike, DICOM datoteka također sadrži i ostale metaoznake vezane uz sami medicinski pregled. Neke metaoznake su obavezne i svaka ih datoteka sadrži, a neke su opcionalne i ovise o vrsti uređaja i tipu

medicinskog pregleda. Svaka datoteka ima **zaglavlje** (engl. **header**) i **skup podataka** (engl. **data set**), kao što je prikazano na slici 2.2. Skup podataka predstavlja instancu jednog informacijskog skupa iz stvarnog svijeta koji se dodatno sastoji od manjih elemenata, nazvanih **podatkovna jedinica** (engl. **data element**).[8]



Slika 2.2 Prikaz DICOM datoteke

Podatkovne jedinice sastoje se od polja. Tri su polja zajednička za sve podatkovne jedinice: oznaka podatkovne jedinice, duljina vrijednosti i polje vrijednosti. Četvrto polje, reprezentacija vrijednosti polja, prisutno je samo u nekim eksplicitnim tipovima podatkovnih jedinica. Strukturu podatkovne jedinice možemo vidjeti na slici 2.3.



Slika 2.3 Prikaz strukture podatkovne jedinice

**Oznaka podatkovne jedinice** (engl. **Data Element Tag**) jest uređeni par 16-bitnih pozitivnih cijelih brojeva koji predstavlja broj grupe metaoznaka i broj elementa unutar grupe. Uređen je u formatu "gggg.eeee", gdje slovo "g" predstavlja broj grupe, a slovo "e" element te grupe.

**Reprezentacija vrijednosti polja** (engl. **Value Representation**) sastoji se od dva jednobajtna znaka. Ovo polje predstavlja tip i oblik podatka koji se nalazi u polju vrijednosti. Polje je prisutno samo u dvije eksplicitne strukture podatkovnih jedinica.

**Duljina vrijednosti** (engl. **Value Length**) jest 16-bitni ili 32-bitni pozitivni cijeli broj koji označava duljinu podatka zapisanog u polju vrijednosti.

**Polje vrijednosti** (engl. **Value Field**) sadrži paran broj bajtova koji označavaju jednu ili više vrijednosti podatkovne jedinice.

Neka medicinska snimanja mogu zapisivati više od jedne slike u seriju za potrebe medicinskog pregleda. Neka snimanja sadrže čak i nekoliko serija snimanja.

## *Poglavlje 2. PACS sustav*

Upravo radi toga, svaka medicinska slika ima više različitih metaoznaka koje koriste za ispravnu identifikaciju u PACS sustavu:

**Study Instance UID** (0020,000D) je metaoznaka koja služi kao jedinstveni identifikator medicinskog pregleda.

**Series Instance UID** (0020,000E) je metaoznaka koja služi kao jedinstveni identifikator serije unutar jednog medicinskog pregleda.

**Instance Number** (0020,0013) je metaoznaka koja služi kao identifikator slike unutar serije.

# Poglavlje 3

## Podatci

Podatci na kojima je razvijeno pretraživanje slika temeljeno na njihovom sadržaju nalaze se na računalu fakulteta, točnije na radnoj stanici za znanstveno računanje "Bonaca". U ovom poglavlju opisat će se njihovo podrijetlo, način pohrane i način obrade.

### 3.1 Podrijetlo podataka

Na radnoj stanici sakupljeni su podatci iz PACS sustava Kliničkog bolničkog centra Rijeka u DICOM formatu. Skup sadrži 29,815,483 medicinskih slika i pripadajućih metaoznaka te je nastao u razdoblju između 2001. i 2016. godine [6].

Zatim, izvršene su razne obrade navedenog skupa podataka te je nastalo nekoliko skupova podataka koji imaju reducirani broj podataka iz DICOM formata [7]. Za potrebe ovog rada korištene su dva takva skupa podataka. Prvi skup sadrži samo matricu intenziteta piksela od 8,000 slika. Matrica intenziteta je preuzeta iz metaoznake Pixel Data (7FE0,0010) te je smanjena veličina slika na 256x256 piksela (većina medicinskih slika u originalu sadrži 512x512 piksela). Drugi skup sadrži samo 15 odabranih metaoznaka vezano za svaku od tih 8,000 slika. Metaoznake važne za ovaj rad su Modality (0008,0060) koja označava modalitet snimanja medicinskog pregleda (npr. CT, MRI) i BodyPartExaminedFixed, koja označava dio tijela koji je trenutno snimljen za potrebe medicinskog pregleda. Metaoznaka BodyPartExaminedFixed



dodana je u procesu grupiranja slika iz DICOM datoteka sa ciljem popunjavanja nepotpunih vrijednosti iz metaoznake BodyPartExamined (0018,0015).

## **3.2 Način pohrane podataka**

Izvorni skup podataka u DICOM formatu sadrži oko 8 terabajta podataka i sastoji se od medicinskih slika i popratnih metaoznaka, koje su zapisane automatski od strane uređaja koji je obavljao pregled ili ručno od strane medicinskog osoblja. Takav skup podataka je nepogodan za obradu zbog ograničenosti resursa radne stanice. Trenutno nije moguće učitati sve podatke u radnu memoriju i vršiti operacije potrebne za analizu cijelog skupa. Iz tog razloga, zanemarene su slike koje nisu sadržavale metaoznake te je izvršena podjela izvornog skupa na nekoliko podskupova [7].

Kao što je navedeno u prošlom potpoglavlju, za ovaj rad korištena su dva skupa podataka. Prvi skup podataka, koji predstavlja matricu intenziteta piksela za 8,000 slika, zapisan je kao numpy array u datoteci `test_images_matrix_single.npy`. Dimenzija numpy array-a je 8000x1x256x256. Ovaj skup podataka je podskup većeg skupa podataka istog formata koji sadrži 30,000 slika u datoteci pod nazivom `images_matrix_single.npy`. Drugi skup podataka, koji sadrži 15 odabranih metaoznaka vezanih za tih 8,000 slika, zapisan je kao format "pandas.DataFrame" u pickle datoteci pod imenom `df_paper_fixed_test_stage _2`. Dimenzije "pandas.DataFrame" formata su 8000x15. Ovaj skup podataka je podskup većeg skupa podataka istog formata koji sadrži metaoznake za 30,000 slika u datoteci pod nazivom `df_paper_fixed_test_stage _2`. Navedeni skupovi podataka su detaljnije opisani u tablici 3.1.

| Skup podataka  | Količina slika | Veličina datoteke |
|--|----------------|-------------------|
| Izvorni skup DICOM datoteka                                    | 29.82M         | 8 TB              |
| Matrica intenziteta piksela<br>(images_matrix_single.npy)      | 30K            | 14.6 GB           |
| Matrica intenziteta piksela<br>(test_images_matrix_single.npy) | 8K             | 1.95 GB           |
| Reducirane metaoznake<br>(df_paper_fixed_stage _2)             | 30K            | 3.65 MB           |
| Reducirane metaoznake<br>(df_paper_fixed_test_stage _2)        | 8K             | 970 KB            |

Tablica 3.1 Prikaz detalja korištenih skupova podataka

### 3.3 Oprema i alati korišteni u radu

U ovom poglavlju bit će opisani računalna oprema i alati koji su bili korišteni u izradi ovog rada.

#### 3.3.1 Računalna oprema

Osobna računala nemaju dovoljne računalne resurse za obradu ovakvog broja podataka u nekom normalnom vremenskom roku, pa iz tog razloga su operacije na podacima važne za ovaj rad izvršene na radnoj stanici "Bonaca". Glavne komponente hardvera čine:

**Procesor** (CPU): 2 x Intel Xeon E5-2620 v4 (8 jezgri i 16 dretvi)

**Grafička kartica** (GPU): 3 x NVIDIA GeForce RTX 2080 Ti

**Radna memorija** (RAM): 128 GB

#### 3.3.2 Programski jezik i paketi

Za potrebe ovog rada korišten je programski jezik Python. Od programskih paketa unutar Python-a korištene su knjižnice: **pandas** i **pickle** za analizu i manipulaciju podacima, **opencv-python** za razne operacije nad slikama, **tensorflow** i **keras**

### *Poglavlje 3. Podatci*

za treniranje i analizu modela neuronskih mreža, **scikit-learn** (sklearn) za traženje najbližih susjeda, **numpy** za rad s matricama (puno knjižnica koriste numpy array za svoj rad), **tkinter** i **Pillow** za grafičko sučelje aplikacije i **pydicom** za čitanje i rad sa DICOM datotekama. Također, za potrebe ovog rada napravljena je knjižnica **defaultValues** koja sadrži određene postavke potrebne za rad cijele aplikacije, poput lokacije datoteka na disku, početnih vrijednosti varijabli i slično.

## Poglavlje 4

# Pretraživanje slika PACS sustava prema njihovom sadržaju

Pretraživanje slika (tražilica slika, engl. image search engine) je primjena tehnika unutar računalnog vida (područje umjetne inteligencije) koja služi za pretraživanje, dohvaćanje i pregledavanje slika u velikim bazama podataka digitalnih slika. Za traženje slika, korisnik može unijeti pojmove upita kao što su ključna riječ, slikovna datoteka/link ili kliknuti na neku sliku, a tražilica će vratiti slike "slične" upitu. Sličnost koja se koristi za kriterije pretraživanja mogu biti metaoznake, raspodjela boja u slikama, atributi oblika ili lokacije unutar slike i slično [9][10].

### 4.1 Tehnike pretraživanja slika

Istraživači i znanstvenici razvili su mnoge tehnike za pronalaženje slika. U ovome poglavlju opisane su neke od najvažnijih i najčešće korištenih tehnika pronalaženja slika. Prva je **pretraživanje pomoću metaoznaka**, također poznata i kao **tekstualno pronalaženje slika** (engl. text-based image retrieval, description-based image retrieval). Većina metoda pretraživanja slika koriste dodavanje neke vrste metaoznaka slikama u bazi podataka kao parametar po kojima tražilica može pronaći rezultate, kao što su ključne riječi, naslov, anotacije ili opisi slika. Ručno označavanje slike je dugotrajno, naporno i skupo, kako bi se to riješilo, provedena je velika

#### *Poglavlje 4. Pretraživanje slika PACS sustava prema njihovom sadržaju*

količina istraživanja o automatskom označavanju slika. Dodatno, porast društvenih web aplikacija pokrenuo je razvoj nekoliko web alata za označavanje slika. Neke od poznatih tražilica su Google, Bing!, i Yahoo.

Druga vrsta pretrage slika jest **pretraživanje po sadržaju** (često nazvan **CBIR**, engl. content-based image retrieval). Tražilice ove vrste analiziraju sadržaj same slike te pretražuju i dohvaćaju na temelju sličnosti njihovog vizualnog sadržaja koristeći značajke te slike, a ne metaoznake kao što su ključne riječi, anotacije, naslovi ili opisi slika. Izraz "sadržaj" u ovom kontekstu može se odnositi na boje, oblike, teksture ili bilo koje druge informacije koje se mogu izvesti iz same slike. CBIR je često poželjan način pretraživanja jer pretraživanja koja se oslanjaju isključivo na metaoznake ovise o kvaliteti i potpunosti zabilješki koje mogu biti zapisane ručno ili automatski. Procjena učinkovitosti pretraživanja slika po ključnim riječima subjektivna je i nije dobro definirana, ljudi mogu imati različitu percepciju i shvaćanje sadržaja slike. Neke od poznatih i javno dostupnih tražilica su TinEye, Pixolution, Picalike.

Treća vrste pretrage je **multimodalno fuzijsko pretraživanje slika** (engl. multimodal fusion image retrieval) koje uključuje spajanje (fuziju) podataka i algoritme strojnog učenja. Fuzija podataka, također poznata kao kombinacija dokaza, tehnika je spajanja više izvora dokaza.

Četvrta vrsta pretrage je **semantičko pretraživanje slika** (engl. semantic-based image retrieval) koje trenutno istražuju mnogi istraživači. Ovo je jedan od pokušaja rješavanja problema semantičkog jaza (engl. semantic gap), problema koji u računarstvu označava prenošenje ljudskih opažanja, opisivanja, aktivnosti i zadataka u računalni svijet, programske jezike i alate. U tom području postoje dva glavna pristupa: označavanje slika ili dijelova (segmenata) slika s ključnim riječima kroz automatsko označavanje slika ili usvajanje semantičkih web inicijativa.

Peta vrsta pretrage koja će biti spomenuta u ovom poglavlju jest **pretraživanje slika pomoću povratnih informacija o relevantnosti** (engl. relevance feedback image retrieval). Ograničena točnost dohvaćanja slika u tražilicama jest u biti posljedica intrinzičnog semantičkog jaza. Osnovna ideja povratne informacije o relevantnosti je integrirati subjektivnost ljudske percepcije u upit pretrage i uključiti korisnika u procjenu rezultata pretraživanja. Tim putem se procjene sličnosti koje se koriste za pretragu prilagođavaju i usavršavaju [11].

## 4.2 Problem pretraživanja voluminoznih podataka PACS sustava

Jedna od mogućih tehnika pretraživanja slika PACS sustava je pretraživanje pomoću metaoznaka. Kao što je već navedeno, dio metaoznaka medicinskih slika u DICOM formatu automatski unosi uređaj koji se koristi za snimanje, a drugi dio ručno unosi medicinsko osoblje. Ovaj postupak ručnog označavanja čovjeka ima potencijal ponuditi vrlo precizne informacije kada su slike dobro označene. Međutim, ovaj proces postaje puno teži za izvođenje kako se veličina zbirke slika povećava, stoga postaje radno intenzivan, dugotrajan i najčešće nemoguć u okvirima nekih realnih vremenskih rokova. Osim toga, neke slike ne mogu biti ispravno označene jer sadrže previše informacija koje bi mogle biti unesene u jednu metaoznaku. Još jedna poteškoća jest subjektivnost ljudske percepcije tijekom označavanja. U prevladavanju problema pretraživanja pomoću metaoznaka, istraživači su pomaknuli fokus na **pretraživanje slika po sadržaju (CBIR)**.

Dvije glavne komponente u ovoj tehnici su **ekstrakcija značajki** (izdvajanje značajki, engl. feature extraction) i **redukcija značajki** (smanjenje značajki, engl. feature reduction) uz primjenu određenog mjerenja sličnosti slika. Sadržaj slike uključuje boju, teksturu i oblik koji se izdvajaju za kasniju upotrebu tijekom procesa indeksiranja i pronalaženja. Takve se značajke smatraju niskom razinom (engl. low-level form) te se obično prikazuju u numeričkom obliku i zapisuju kao **vektor značajki** (engl. feature vector). Većinu korisnika često zanima samo određeni dio (regija) slike, a ne cijela slika. Stoga većina trenutnih CBIR zasnovana su na regijama, gdje se značajke izvlače samo iz regija interesa (engl. regions of interest). Predstavljanje slika na regionalnoj razini često se smatra kao bliže ljudskoj percepciji.

U prošlosti se ekstrakcija značajki niske razine vršila raznim algoritmima, nakon kojih bi uslijedio proces redukcije značajki kako bi se formirao dobar prikaz značajki iz većeg skupa značajki. Međutim, ti algoritmi nisu u stanju adekvatno predstaviti semantičke informacije ili značajke (koncepte) visoke razine, uključujući ključne riječi i tekstualne opise koje ljudi obično koriste tijekom tumačenja. Algoritmi su djelomično ograničeni kada se radi o većem broju slika bogatog sadržaja u bazi podataka. Istraživači su, u prevladavanju ovih problema, ponudili nekoliko rješenja

za smanjenje semantičkog jaza u pretraživanju slika. Jedno od njih je korištenje povratnih informacija o relevantnosti, koje je već opisano u ovom poglavlju. Tim načinom se povećava preciznost rezultata pretrage tako što svaki dobiveni rezultat tražilice korisnik ocjenjuje, a sustav uzima tu informaciju te prilagođava daljnje pretrage. Također, često rješenje je korištenje alata strojnog učenja u pretraživanju, poput implementiranja neuronskih mreža za ekstrakciju značajki ili povezivanja značajki niske razine sa konceptima upita (koncepti više razine, područje ontologije u računarstvu).

Osim ekstrakcije i redukcije značajki, još jedan ključni izazov u pretraživanju po sadržaju je **mjerenje sličnosti**. Izračunavanjem udaljenosti između odgovarajućih vektora značajki (često ih se naziva i deskriptorima značajki, engl. feature descriptors) u zadanom prostoru, može se odrediti sličnost između bilo koje dvije slike. Postoje različiti načini za mjerenje sličnosti značajki slike, npr. pomoću metrike udaljenosti kao što su euklidska udaljenost, Manhattan, Minkowski ili Hamming udaljenost, gdje su značajke predstavljene kao vektor ili putem podudaranja grafova gdje su objekti na slikama raspoređeni u obliku odnosa među sobom. Osim toga, statistički klasifikatori (engl. statistical classifiers) također se mogu trenirati za kategoriziranje slike upita u odgovarajuće zadane klase [12].

## 4.3 Prototip rješenja

Iz razloga koji su navedeni u ovome poglavlju, ovaj rad koristit će pretraživanje slika po sadržaju za potrebe razvijanja prototipa rješenja. Ono se sastoji od nekoliko koraka. Prvi korak je **indeksiranje baze podataka**, tj. ekstrakcija značajki svih slika zapisanih u zadanom skupu podataka i pohranjivanje dobivenih vektora značajki u određenom obliku kako bi se naknadno mogli koristiti za mjerenje sličnosti. U idućem koraku vrši se ekstrakcija značajki iz slike zadane unosom korisnika kojom dobijemo vektor značajki. Zatim vršimo **mjerenje sličnosti**, dobiveni vektor značajki uspoređujemo sa svim vektorima značajki zapisanima u procesu indeksiranja baze podataka te dobijemo ocjenu sličnosti za svaki vektor značajki. Ocjena sličnosti je u obliku  $d = [0, +\infty >$  gdje vrijednost 0 označava da su slike identične, a što su vrijednosti veće od 0, to su slike manje slične jedna drugoj. Nadalje, potrebno je

#### *Poglavlje 4. Pretraživanje slika PACS sustava prema njihovom sadržaju*

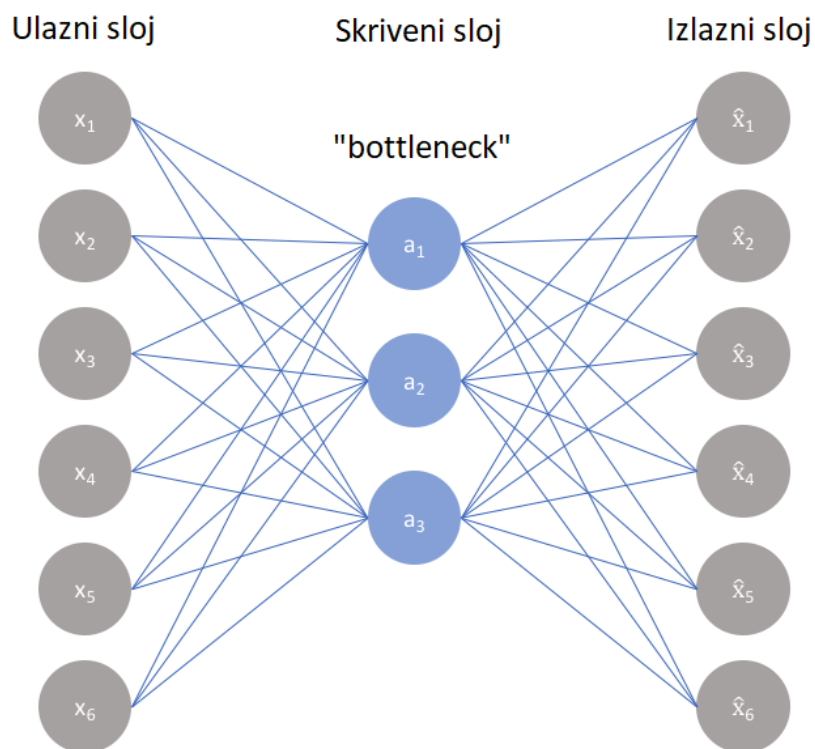
pronaći najsličnije slike u bazi podataka, tj. pronaći slike čiji vektori značajki imaju najmanju ocjenu sličnosti sa vektorom značajki zadane slike. Za tu svrhu koristi se **pronazak najbližih susjeda** (engl. nearest neighbour search), algoritam korišten u optimizaciji za pronalaženje točke u danom skupu koja je najbliža (ili najsličnija) danoj točki.

U prototipu rješenja ovoga rada implementirati ćemo alate strojnog učenja za ekstrakciju značajki slika i za pronazak najbližih susjeda. Za ekstrakciju značajki koristit ćemo **autokoder**, a za pronazak najbližih susjeda upotrijebit ćemo **algoritam k-najbližih susjeda**.

##### **4.3.1 Autokoder**

Autokoder (engl. autoencoder) je vrsta umjetne neuronske mreže koja se koristi u strojnom učenju bez nadzora (engl. unsupervised learning) za učenje učinkovitog predstavljanja podataka, koje se još naziva i kodiranje podataka (engl. data encoding). Mreža autokodera ima tri sloja: **ulazni sloj** (engl. input layer) u kojemu imamo ulazne podatke (sliku koju želimo opisati vektorom značajki), **skriveni sloj** (engl. hidden layer) za kodiranje ulaznog sloja u kompresirani (reducirani) prikaz koji se na engleskom jeziku često naziva "bottleneck" (hrv. usko grlo) i **izlazni sloj** (engl. output layer) za dekodiranje skrivenog sloja [13][14]. Struktura mreže prikazana je na slici 4.1.



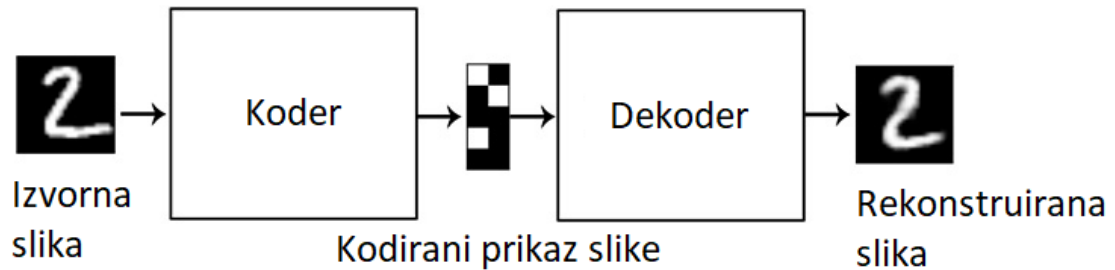


Slika 4.1 Struktura mreže autokodera

Koristeći algoritam povratnog širenja (engl. backpropagation), nenadzirani algoritam kontinuirano se trenira postavljanjem ciljanih izlaznih vrijednosti na vrijednosti jednake ulaznim. Ovo prisiljava manji, skriveni sloj kodiranja da koristi redukciju dimenzija značajki kako bi eliminirao šum, tj. nepotrebne podatke i rekonstruirao ulazne podatke.

Autokoder ima dva glavna dijela: **koder** (engl. encoder) koji preslikava ulaz u kodirani prikaz i **dekoder** (engl. decoder) koji preslikava kodirani prikaz u rekonstrukciju ulaza. Primjer autokodera prikazan je na slici 4.2.

Poglavlje 4. Pretraživanje slika PACS sustava prema njihovom sadržaju



Slika 4.2 *Primjer autokodera*

U skupu podataka kojeg koristimo za ovaj rad, medicinske slike su prikazane u dimenziji 256x256 piksela, što je ukupno 65536 piksela, a svaku sliku ćemo opisati vektorom značajki koji sadrži 256 brojeva. U nastavku opisan je programski kod korišten za implementaciju autokodera.

```
1  # modelTrainer.py
2  import keras
3  import numpy as np
4  import modules.defaultValues as defaultValues
5
6  encoding_dim = 256
7  array_dim = 65536
8  epochs_n = 200
9  max_train_data_size = defaultValues.n_results_to_search_max # 8000
10
11 input_img = keras.Input(shape=(array_dim,))
12 encoded = keras.layers.Dense(encoding_dim, activation='relu')(input_img)
13 decoded = keras.layers.Dense(array_dim, activation='sigmoid')(encoded)
14 autoencoder = keras.Model(input_img, decoded)
```

Kodni isječak 1 *modelTrainer.py* datoteka

Na početku, potrebno je uvesti potrebne knjižnice za rad autokodera (linije 2-4,

#### Poglavlje 4. Pretraživanje slika PACS sustava prema njihovom sadržaju

kodni isječak 1). Korištene knjižnice objašnjene u poglavlju 3.3.2. Zatim zadajemo parametre modela autokodera kojeg ćemo trenirati (linije 6-9, kodni isječak 1). Parametar `encoding_dim` definira broj značajki kojima ćemo opisati svaku sliku, `array_dim` označava ukupan broj piksela u izvornoj slici (256x256 piksela, ukupno 65536 piksela), `epochs_n` označava broj iteracija (epoha) koje će se koristiti za treniranje modela, a parametar `max_train_data_size` zadaje maksimalan broj podataka na kojem ćemo trenirati model. Nakon toga, stvaramo model autokodera sa zadanim parametrima (linije 11-14, kodni isječak 1).

```
1  # modelTrainer.py
2  x_train = np.load(defaultValues.default_database_npy_file)
3  x_train = x_train[0:max_train_data_size]
4  array_size = len(x_train)
5
6  x_train = x_train.reshape((len(x_train), np.prod(x_train.shape[1:])))
7  x_test = x_train
8
9  autoencoder.compile(optimizer='adam', loss='binary_crossentropy')
10 autoencoder.fit(x_train, x_train, epochs=epochs_n, batch_size=256,
    ↪ shuffle=True, validation_data=(x_test, x_test))
11 model_to_save = "models/" + str(encoding_dim) + "_" + str(epochs_n) + "_"
    ↪ + str(array_size)
12 autoencoder.save(model_to_save)
```

#### Kodni isječak 2 *modelTrainer.py* datoteka

U idućem koraku zadajemo skup podataka za treniranje modela `x_train` iz datoteke na disku (linija 2, kodni isječak 2) te ga limitiramo na samo 8000 slika radi uštede korištenja resursa računala (linija 3, kodni isječak 2). Za rad modela, potrebno je oblikovati matricu intenziteta piksela tako da svaka slika bude zapisana kao jednodimenzionalno polje u kojem su zapisani sve vrijednosti piksela, umjesto kao dvodimenzionalno polje dimenzija 256x256 (linija 6, kodni isječak 2). Finalna dimenzija matrice skupa podataka jest 8000x65536, umjesto 8000x256x256. Također, potreban je testni skup podataka kojim provjeravamo istrenirani skup podataka

te ga zapisujemo u varijablu `x_test` i trenutno je identičan varijabli `x_train` (linija 7, kodni isječak 2). Sljedeći korak je dodatno konfiguriranje modela (linija 9, kodni isječak 2) gdje zadajemo način optimizatora ove funkcije (`optimizer='adam'`) i način računanja gubitka vrijednosti autokodera (`loss='binary_crossentropy'`). Nakon toga treniramo model uz zadane parametre i dva skupa podataka, istrenirani skup i testni skup (linija 10, kodni isječak 2). Na kraju potrebno je istrenirani model pohraniti kako bi se on mogao koristiti za ekstrakciju značajki slika.

### **4.3.2 Algoritam k-najbližih susjeda**

Za implementaciju algoritma k-najbližih susjeda u ovome radu koristit ćemo "Unsupervised Nearest Neighbors" unutar knjižnice scikit-learn, koji koristi strojno učenje bez nadzora za pronalazak najbližih susjeda. Za izračunavanje udaljenosti, u pravilu možemo koristiti bilo koju metriku udaljenosti, no euklidska udaljenost je najčešći izbor metrike u ovome algoritmu. Metode pretraživanja susjeda poznate su kao nengeneralizirajuće metode strojnog učenja, jer jednostavno "pamte" sve svoje podatke o obuci [15]. U nastavku opisan je programski kod korišten za implementaciju algoritma k-najbližih susjeda.

#### Poglavlje 4. Pretraživanje slika PACS sustava prema njihovom sadržaju

```
1  # databaseIndexer.py
2  import keras
3  import numpy as np
4  from sklearn.neighbors import NearestNeighbors
5  import pickle
6  import modules.defaultValues as defaultValues
7
8  max_results_to_search = defaultValues.n_results_to_search_max
9
10 database = np.load(defaultValues.default_database_npy_file)
11 database = database[0:max_results_to_search]
12
13 database = database.reshape((len(database), np.prod(database.shape[1:])))
14
15 autoencoder =
    ↪ keras.models.load_model(defaultValues.default_autoencoder_model_name)
16 encoder = keras.Model(autoencoder.input, autoencoder.layers[-2].output)
```

Kodni isječak 3 *databaseIndexer.py* datoteka

Na početku uvodimo potrebne knjižnice (linije 1-6, kodni isječak 3). U poglavlju 3.3.2 navedene su i objašnjene sve potrebne knjižnice za rad aplikacije. Zatim učitavamo skup podataka za indeksiranje baze podataka u varijablu `database` (linija 10, kodni isječak 3) i limitiramo ga na 8000 slika radi uštede resursa računala (linije 8 i 11, kodni isječak 3). Kao što je već navedeno u opisu rada autokodera, za korištenje modela autokodera potrebno je preoblikovati matricu intenziteta piksela u dvodimenzionalno polje dimenzija 8000x65536 (linija 13, kodni isječak 3). Zatim, učitavamo model autokodera kojeg smo istrenirali u poglavlju 4.3.1 (linija 15, kodni isječak 3) te iz njega uzimamo koder, dio modela koji je zadužen za dobivanje kodiranog prikaza slike (linija 16, kodni isječak 3).

#### Poglavlje 4. Pretraživanje slika PACS sustava prema njihovom sadržaju

```
1  # databaseIndexer.py
2  knn = NearestNeighbors(n_neighbors=20, metric="minkowski")
3
4  encoded_imgs = encoder.predict(database)
5
6  knn.fit(encoded_imgs)
7  knn_pickle = open(defaultValues.default_knn_model_name, "wb")
8  pickle.dump(knn, knn_pickle)
9  knn_pickle.close()
```

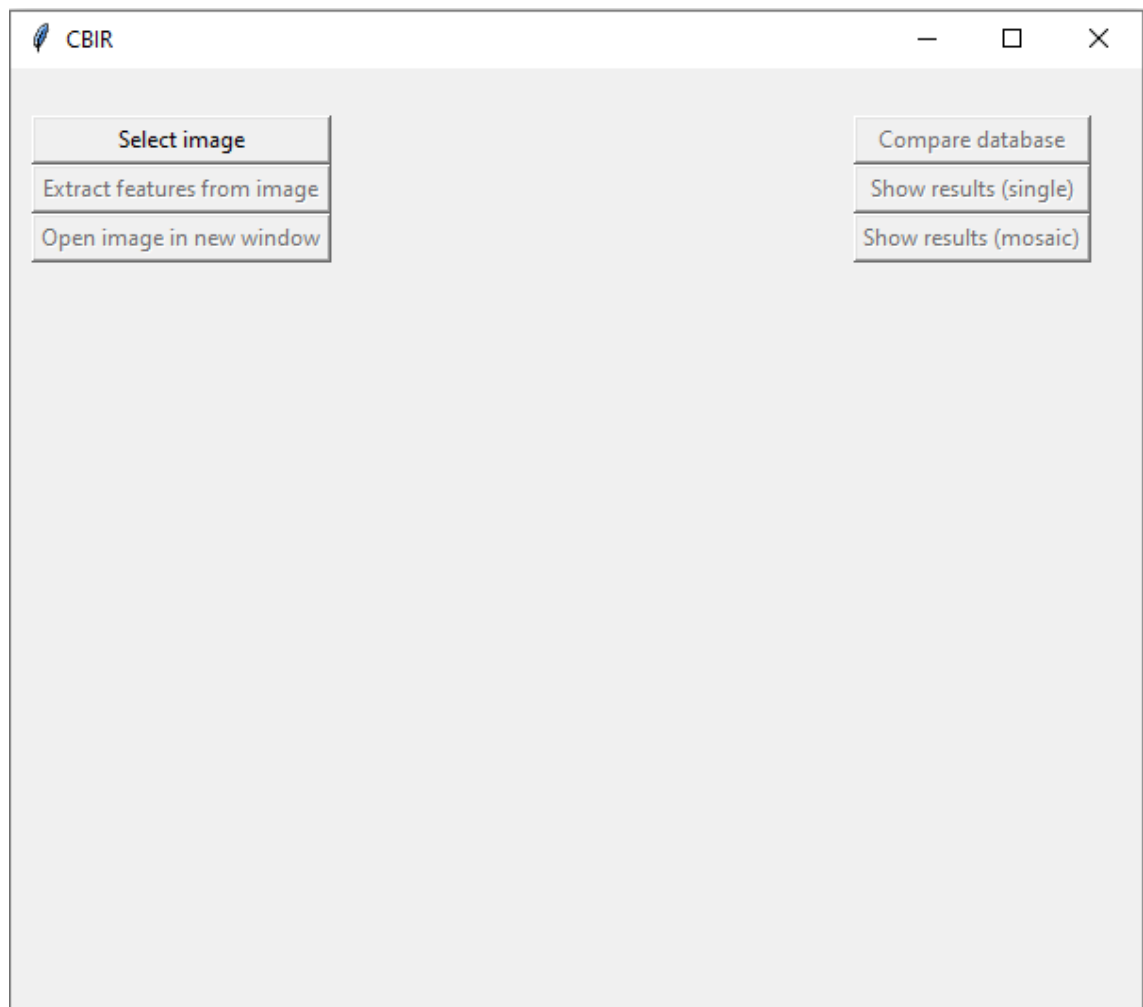
Kodni isječak 4 *databaseIndexer.py* datoteka

Idući korak je konfiguriranje modela učenja pronalaska najbližih susjeda (linija 2, kodni isječak 4). Za broj najbližih susjeda  $k$  zadajemo vrijednost 20 koju smo sami odredili za demonstraciju rješenja, a za metriku udaljenosti koristimo Minkowskijevu udaljenost koja je generalizacija euklidske i Manhattan udaljenosti. Nakon toga, indeksiramo bazu podataka tako što koristimo koder (dobiven iz istreniranog modela autokodera) za ekstrakciju značajki i dobivene vektore zapišemo u varijablu `encoded_imgs` (linija 4, kodni isječak 4). Sada možemo istrenirati model pronalaska susjeda tako što koristimo varijablu `encoded_imgs` kao trenirani skup podataka modela (linija 6, kodni isječak 4). Trenirani model potrebno je pohraniti za kasnije korištenje u pretraživanju najbližih susjeda (linije 7-9, kodni isječak 4).

### 4.3.3 Tražilica

Za demonstraciju rada prototipa rješenja napravljeno je grafičko korisničko sučelje pomoću `tkinter` i `Pillow` knjižnica unutar programskog jezika Python. Na slici 4.3 možemo vidjeti izgled sučelja.

Poglavlje 4. Pretraživanje slika PACS sustava prema njihovom sadržaju



Slika 4.3 Grafičko sučelje *CBIR.py* datoteke

Za potrebe testiranja i razvijanja ovog CBIR sustava, manji skup slika je pohranjeno u formatu numpy array u zasebne datoteke. Tipka "Select image" otvara sučelje za biranje datoteke koja sadrži sliku koju želimo koristiti kao parametar pretraživanja PACS sustava po sadržaju. Tipka "Open image in new window" otvara novi prozor u kojemu je prikazana slika odabrane datoteke. Pritiskom na tipku "Extract features from image" vršimo ekstrakciju značajki slike unutar odabrane datoteke pomoću treniranog modela autokodera. Ključan dio ovog koraka je da koristimo isti model autokodera kojeg smo koristili za indeksiranje baze podataka, inače rezultati neće biti ispravni. Tipka "Compare database" pokreće proces uspoređivanja

#### Poglavlje 4. Pretraživanje slika PACS sustava prema njihovom sadržaju

značajki odabrane slike sa značajkama svih slika u bazi podataka dobivenih indeksiranjem baze podataka te pohranjuje 20 najslićnijih slika u varijablu koja se može kasnije koristiti za prikaz rezultata pretrage. Tipke "Show results (single)" i "Show results (mosaic)" otvaraju novi prozor koji prikazuje 20 najslićnijih slika. Prva tipka prikazuje tih 20 slika kao zasebne slike, dok druga tipka prikazuje svih 20 slika kao mozaik. Programski kod korišten za tražilicu prikazan je u nastavku.

```
1  # CBIR.py
2  autoencoder =
   ↪  keras.models.load_model(defaultValues.default_autoencoder_model_name)
3  encoder = keras.Model(autoencoder.input, autoencoder.layers[-2].output)
4
5  features_to_search = encoder.predict(image_selected)
6
7  knnPickle = open(defaultValues.default_knn_model_name, "rb")
8  knn = pickle.load(knnPickle)
9  knnPickle.close()
10
11 results_distances, results_indexes =
   ↪  knn_model.kneighbors(features_to_search)
12 results_distances = results_distances[0]
13 results_indexes = results_indexes[0]
```

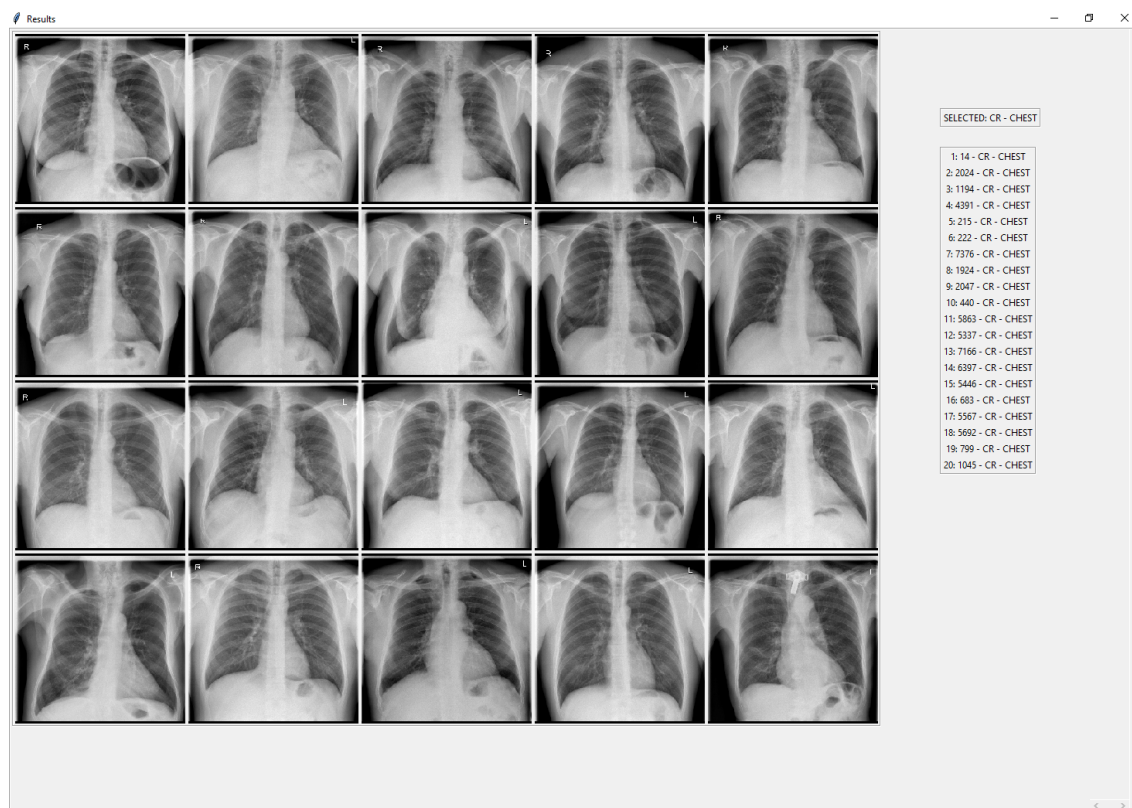
#### Kodni isječak 5 *CBIR.py* datoteka

Kao kod postupka indeksiranja baze podataka, potrebno je učitati trenirani model autokodera te iz njega uzeti koder (linije 2-3, kodni isječak 5). Pomoću tog kodera zatim izvučemo značajke slike kako bi mogli sliku usporediti s bazom podataka (linija 5, kodni isječak 5). Idući korak je učitavanje modela treniranog za pronalazak najbližih susjeda u varijablu knn (linije 7-9, kodni isječak 5). Nakon toga pretražujemo najbliže susjede tako što kao parametar funkcije šaljemo izvučene značajke iz odabrane slike (linije 11-13, kodni isječak 5). Najbliži susjedi zapisani su u obliku dva polja, prvo polje je skup ocjena sličnosti svakog susjeda (`results_distances`), dok je drugo polje skup identifikatora svakog susjeda. U ovom prototipu rješenja,



#### Poglavlje 4. Pretraživanje slika PACS sustava prema njihovom sadržaju

identifikator je broj retka matrice intenziteta piksela koji sadrži vrijednosti piksela slike tog susjeda (results\_indexes). Navedena polja koriste se kako bi napravili listu svih dobivenih susjeda, a koja sadrži identifikator slike i ocjenu udaljenosti te slike. Tu listu koristimo za prikazivanje rezultata pretrage, što činimo pritiskom na tipku "Show results (mosaic)", koja otvara novi prozor prikazan na slici 4.4.



Slika 4.4 Prozor "Show results (mosaic)"

Uz prikaz slika dobivenih susjeda, s desne strane prikazuju se dvije dodatne informacije o dobivenom susjedu, modalitet i dio tijela snimljenog na toj slici, a kako bi imali bolji uvid u dobivene rezultate pretraživanja. Navedene informacije saznajemo pomoću identifikatora, koristeći metaoznake Modality i BodyPartExaminedFixed iz skupa reduciranih metaoznaka iz datoteke df\_paper\_fixed\_test\_stage\_2. Nekoliko primjera pretraživanja te mjerenje vremena potrebnog za te primjere navesti će se u idućem poglavlju.

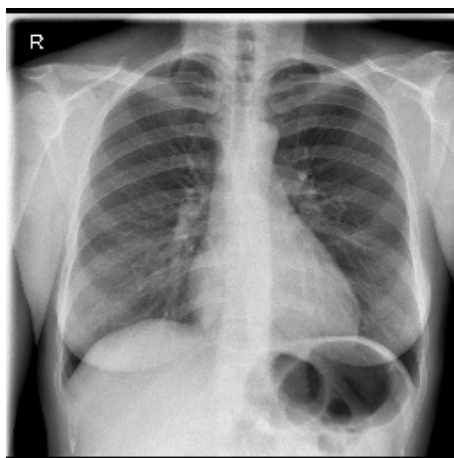
## Poglavlje 5

### Rezultati

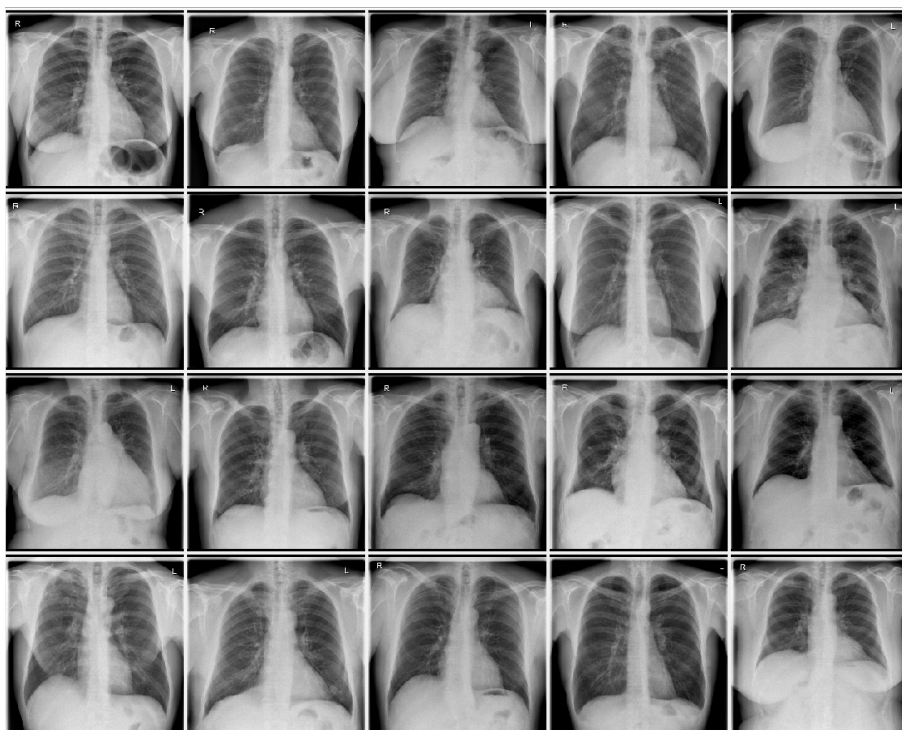
U nastavku prikazat će se primjeri slika korištenih za pretraživanje te dobiveni rezultati. Zatim, prikazat će se mjerenja vremena potrebnog za rad aplikacije prototipa rješenja.

#### 5.1 Rezultati prototipa rješenja

## Poglavlje 5. Rezultati



Slika 5.1 *Primjer 1 - medicinska slika*



Slika 5.2 *Primjer 1 - rezultati*

## Poglavlje 5. Rezultati

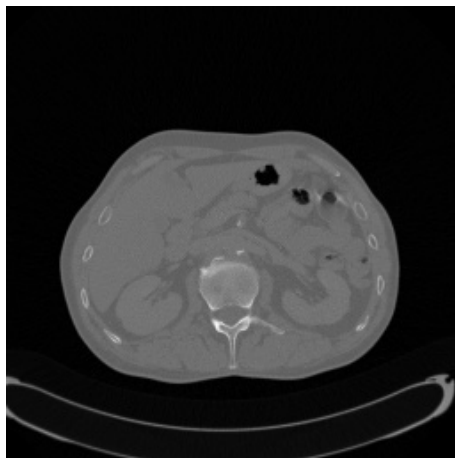


Slika 5.3 *Primjer 2 - medicinska slika*

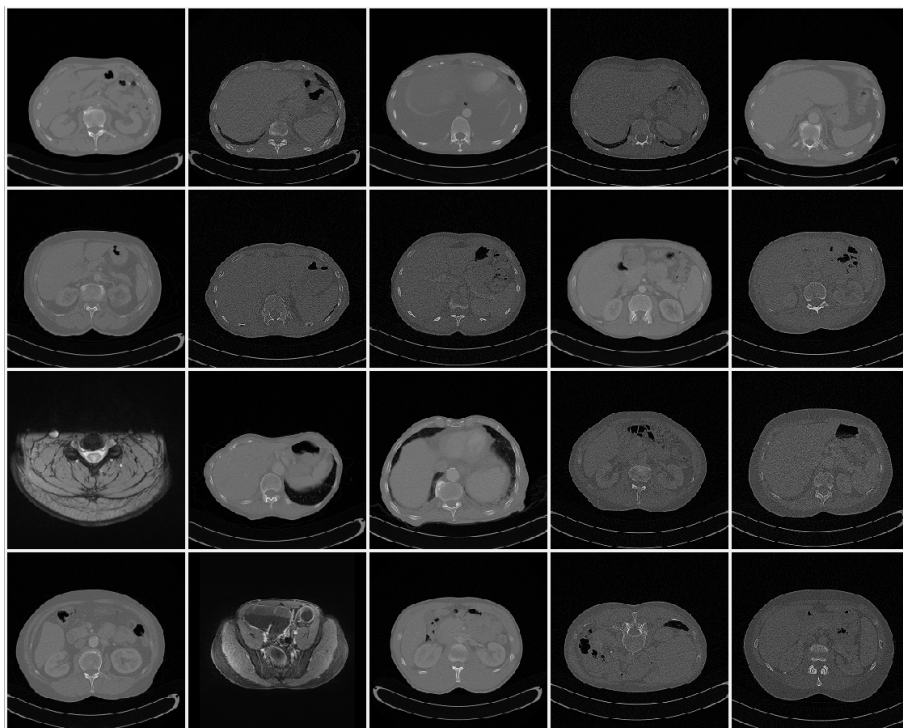


Slika 5.4 *Primjer 2 - rezultati*

## Poglavlje 5. Rezultati



Slika 5.5 *Primjer 3 - medicinska slika*

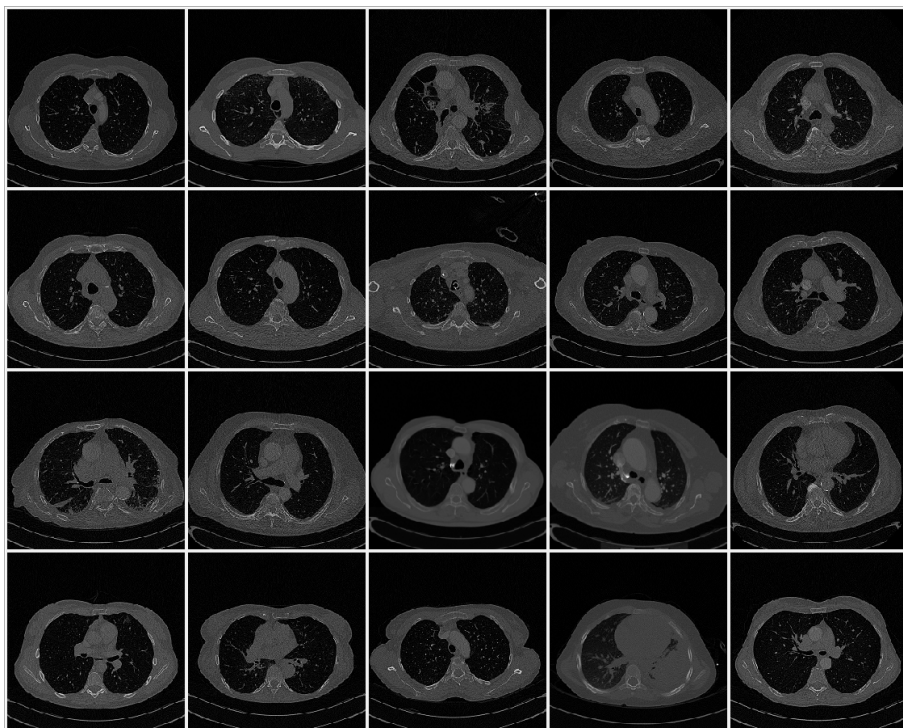


Slika 5.6 *Primjer 3 - rezultati*

## Poglavlje 5. Rezultati

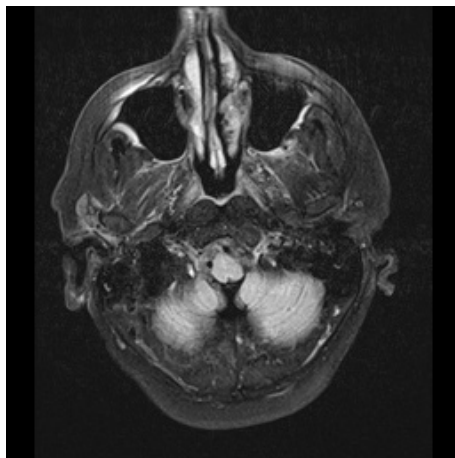


Slika 5.7 *Primjer 4 - medicinska slika*

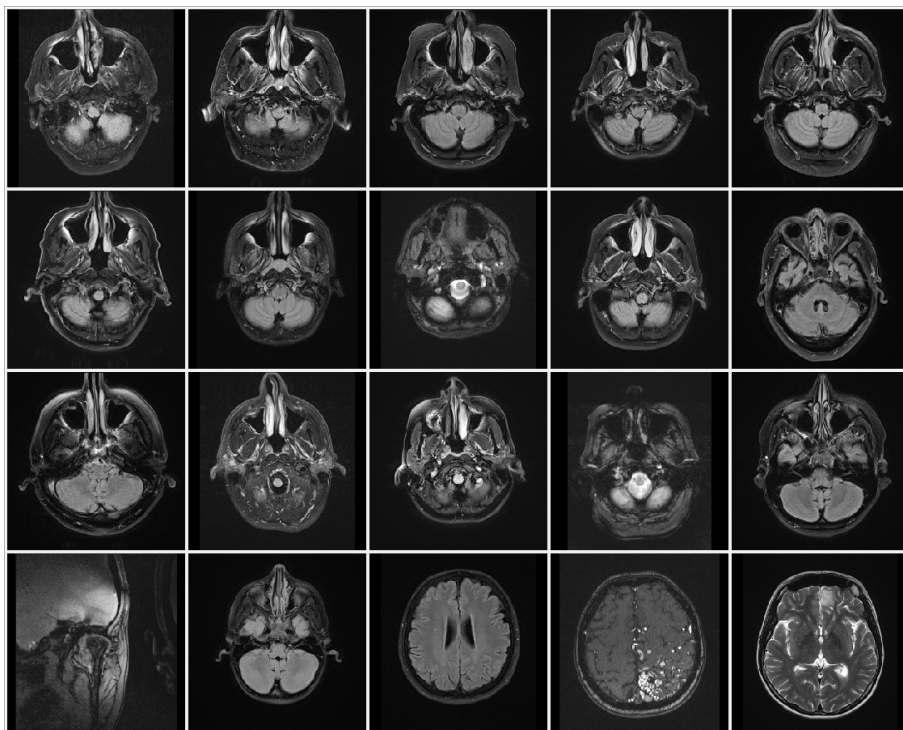


Slika 5.8 *Primjer 4 - rezultati*

## Poglavlje 5. Rezultati

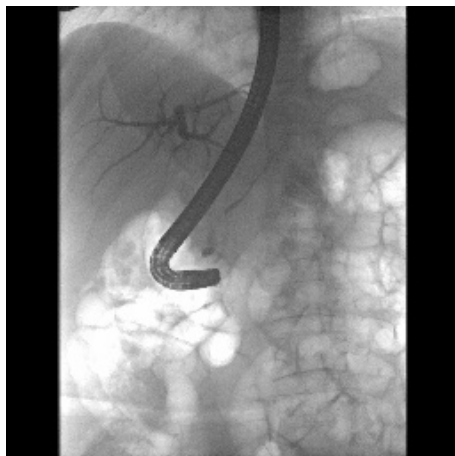


Slika 5.9 *Primjer 5 - medicinska slika*

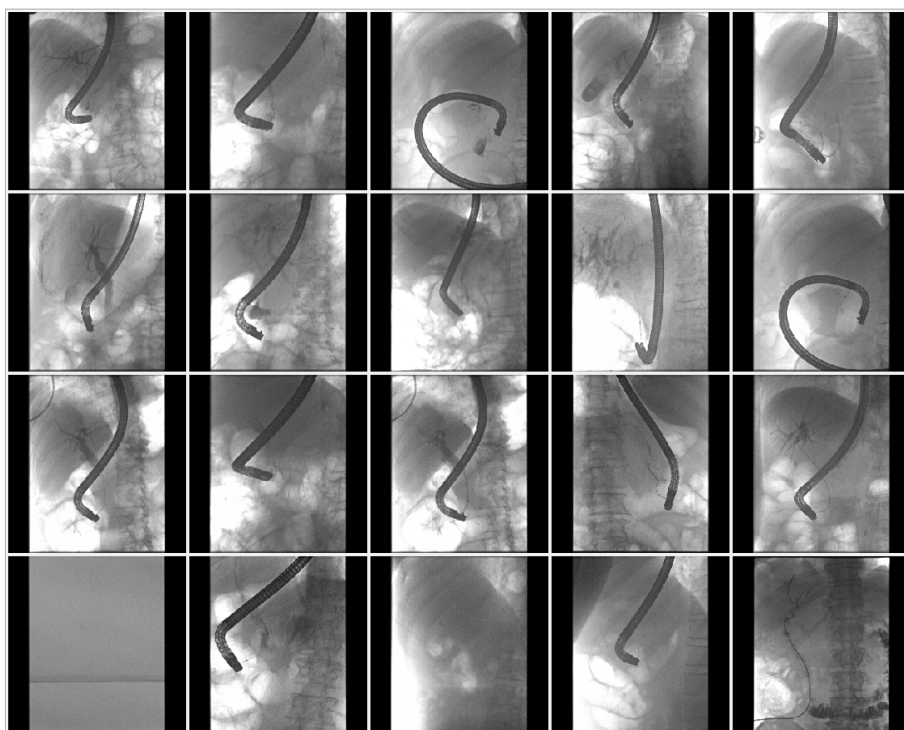


Slika 5.10 *Primjer 5 - rezultati*

## Poglavlje 5. Rezultati



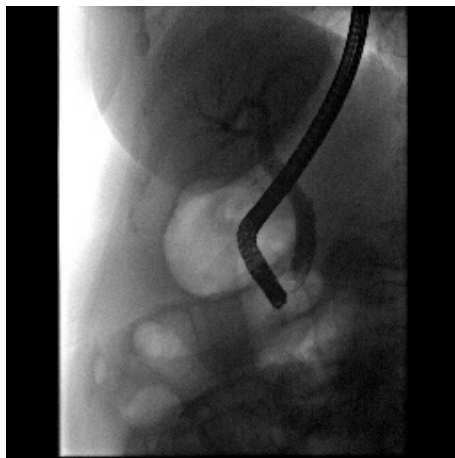
Slika 5.11 *Primjer 6 - medicinska slika*



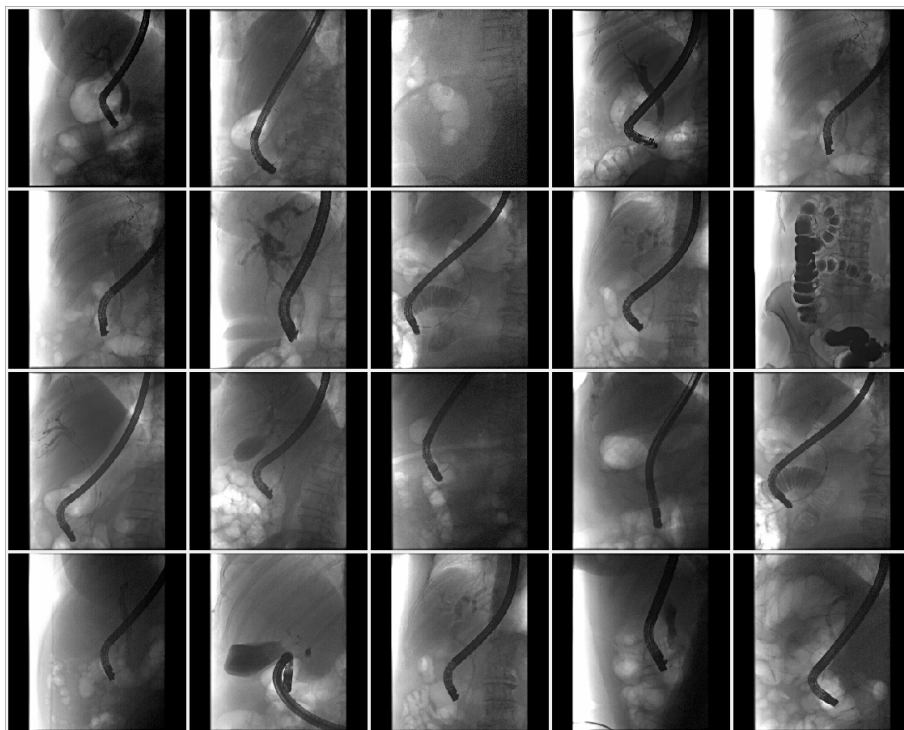
Slika 5.12 *Primjer 6 - rezultati*



Poglavlje 5. Rezultati



Slika 5.13 *Primjer 7 - medicinska slika*

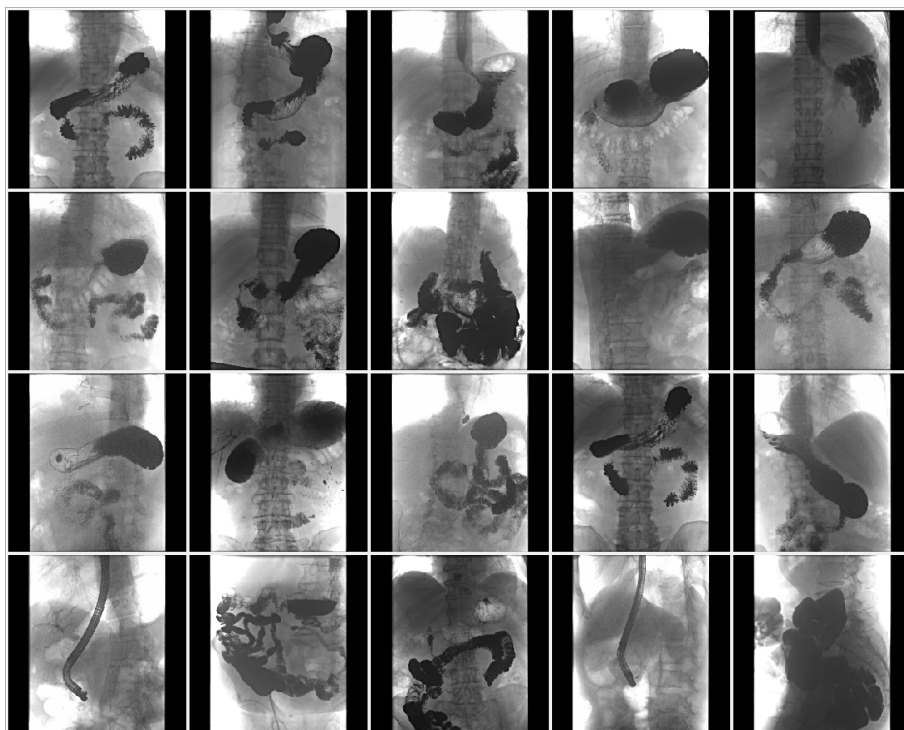


Slika 5.14 *Primjer 7 - rezultati*

Poglavlje 5. Rezultati



Slika 5.15 *Primjer 8 - medicinska slika*



Slika 5.16 *Primjer 8 - rezultati*

## *Poglavlje 5. Rezultati*

Za iduća dva primjera prikazani su i rezultati dobiveni korištenjem kosinusne metrike udaljenosti u algoritmu pretraživanja najbližih susjeda, kako bi se pokazala razlika u rezultatima različitih metrika udaljenosti.

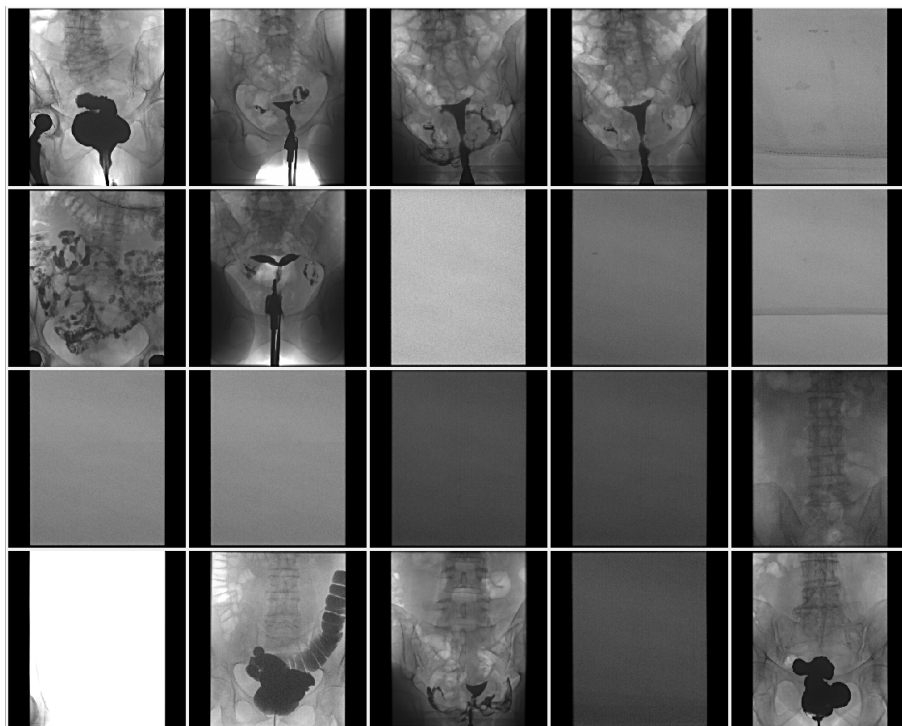


*Slika 5.17 Primjer 9 - medicinska slika*



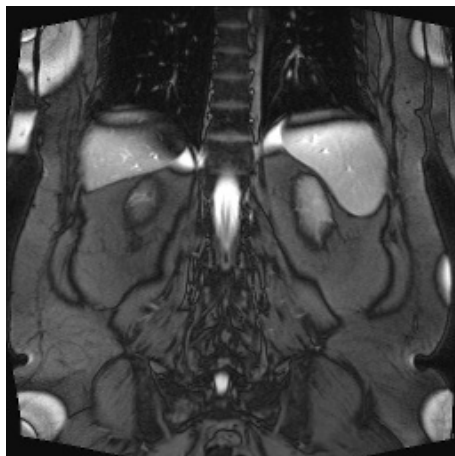
*Slika 5.18 Primjer 9 - rezultati - Minkowskijeva udaljenost*

## Poglavlje 5. Rezultati

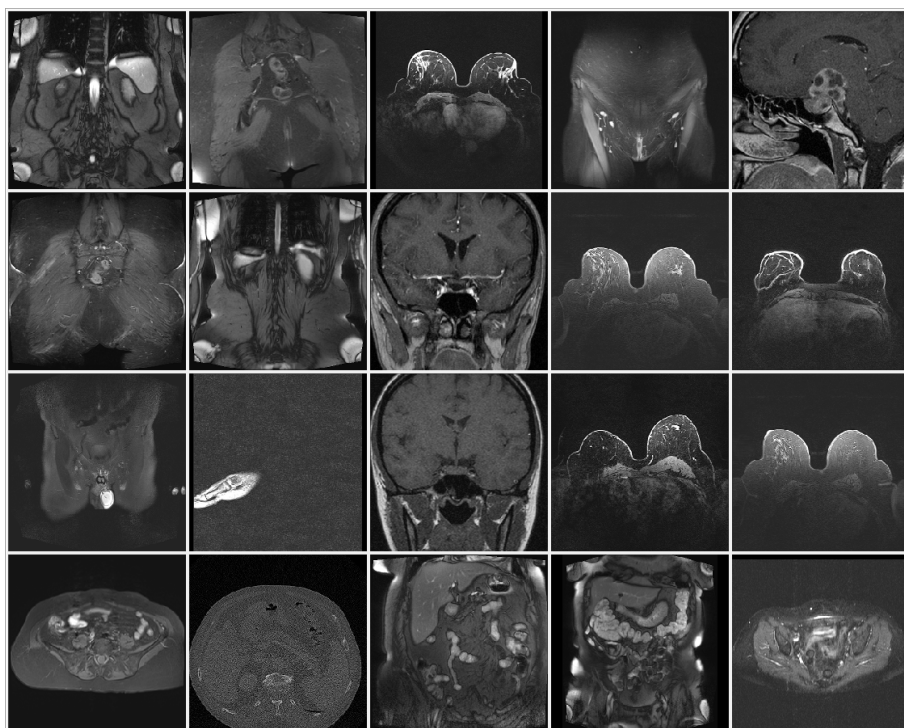


Slika 5.19 *Primjer 9 - rezultati - kosinusna udaljenost*

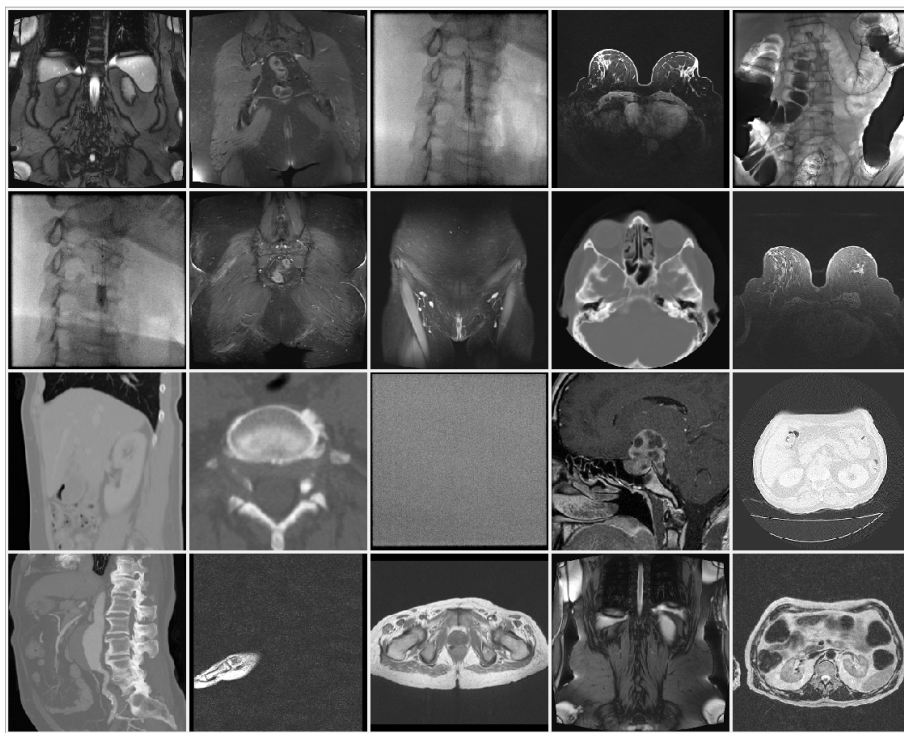
Poglavlje 5. Rezultati



Slika 5.20 *Primjer 10 - medicinska slika*



Slika 5.21 *Primjer 10 - rezultati - Minkowskijeva udaljenost*



Slika 5.22 *Primjer 10 - rezultati - kosinusna udaljenost*

## 5.2 Vremenska analiza

Za treniranje modela autokodera odabran je skup podataka od 8000 slika, 200 epoha (broj iteracija treniranja modela) te 256 značajki koje opisuju svaku sliku. Koristeći računalnu snagu jedne grafičke kartice NVIDIA GeForce RTX 2080 Ti, pomoću knjižnica tensorflow i keras, model se u prosjeku trenira 38 minuta. Za treniranje modela pretraživanja k najbližih susjeda u prosjeku je potrebno 9 sekundi kako bi pronašli 20 najbližih susjeda u skupu od 8000 slika. Vrijeme potrebno za pretraživanje PACS sustava po sadržaju pomoću odabrane slike (iz odabrane datoteke) otprilike iznosi 7 sekundi, od čega je ekstrakcija značajki za tu sliku gotovo instantna (otprilike desetina sekunde) a pronalazak najbližih susjeda u prosjeku traje oko 7 sekundi. Sažetak mjerenja vremena izvršavanje ove aplikacije prikazan je u tablici 5.1.

## Poglavlje 5. Rezultati

| Operacija   | Vremensko trajanje |
|---|--------------------|
| Treniranje modela autokodera                        | $\approx 38$ min   |
| Treniranje modela pretraživanja k najbližih susjeda | $\approx 9$ sek    |
| Pretraživanje PACS sustava pomoću slike             | $\approx 7$ sek    |

Tablica 5.1 Prikaz vremenskog trajanja operacija aplikacije

Kao što možemo vidjeti, najviše vremena (i računalne komputacije) troši se na treniranje modela autokodera. Ono proporcionalno ovisi o broju epoha, broju značajki te broju testnog skupa podataka. Treniranje modela pronalaska najbližih susjeda i pretraživanje PACS sustava pomoću odabrane slike su relativno brze. Ovakvo rješenje vremenski je vrlo učinkovito jer korisnik može pretražiti PACS sustav pomoću odabrane slike u roku od 10 sekundi, treniranje oba modela izvršava se samo jednom prije korištenja aplikacije.

## Poglavlje 6

# Zaključak

Pretraživanje slika PACS sustava prema njihovom sadržaju može pomoći medicinskom osoblju u pronalasku semantički sličnih slika dobivenih medicinskim pretragama, kako bi poboljšali dijagnostiku i liječenje bolesti pacijenata. Također, može pomoći istraživačima i znanstvenicima u radu s podacima vezanima uz medicinu. U ovome radu opisani su PACS sustav i DICOM format pohranjivanja datoteka unutar tog sustava, podatci i alati koji su se koristili za potrebe rada. Nakon toga, dan je pregled pretraživanja slika u području računalnog vida. Opisano je nekoliko tehnika i metoda pretraživanja te problem pretraživanja voluminoznih podataka PACS sustava.

Nakon opisa područja ovog rada ponuđen je prototip rješenja koji koristi strojno učenje za pretraživanje slika, autokoder za ekstrakciju značajki slika te algoritam k-najbližih susjeda za pronalazak najbližih slika po sadržaju. Prikazani su rezultati dobiveni korištenjem razvijene aplikacije nad zadanim podacima i vrijeme potrebno za njeno izvršavanje.

Iako rješenje daje zadovoljavajuće rezultate pretraživanja, ima nekoliko komponenti koje bi se mogle poboljšati, poput korištenja tehnika povratnih informacija o relevantnosti dobivenih rezultata, čime se spaja pretraživanje slika po sadržaju i procjene uspješnosti od strane ljudi, optimiziranja broja značajki izvučenih iz slike za usporedbu te optimiziranja modela za pronalazak najbližih susjeda.



# Bibliografija

- [1] Picture archiving and communication system, s Interneta, [https://en.wikipedia.org/wiki/Picture\\_archiving\\_and\\_communication\\_system](https://en.wikipedia.org/wiki/Picture_archiving_and_communication_system), srpanj 2022.
- [2] Guld, M. O.; Kohnen, M.; Keysers D.; i dr.; "Quality of DICOM header information for image categorization," Proceedings of SPIE - The International Society for Optical Engineering, vol. 4685, 2002.
- [3] DICOM Standard: Introduction and Overview, The Association of Electrical Equipment and Medical Imaging Manufacturers, s Interneta, <https://dicom.nema.org/medical/dicom/current/output/pdf/part01.pdf>, srpanj 2022.
- [4] Digital Imaging and Communications in Medicine, s Interneta, <https://en.wikipedia.org/wiki/DICOM>, srpanj 2022.
- [5] Digital Imaging and Communications in Medicine, s Interneta, <https://www.dicomstandard.org/about-home>, srpanj 2022.
- [6] Burul M.; "Prijenos znanja u analizi medicinskih slika", s Interneta, <https://urn.nsk.hr/urn:nbn:hr:190:594917>, 2017.
- [7] Ilić D.; "Grupiranje sadržaja repozitorija medicinskih slika prema DICOM oznakama" , s Interneta, <https://urn.nsk.hr/urn:nbn:hr:190:021891>, 2019.
- [8] DICOM Standard: Data Structures and Encoding, The Association of Electrical Equipment and Medical Imaging Manufacturers, s Interneta, <https://dicom.nema.org/medical/dicom/current/output/pdf/part05.pdf>, srpanj 2022.
- [9] Datta, R.; Joshi, D.; Li, J.; Wang, J. Z.; "Image Retrieval: Ideas, Influences, and Trends of the New Age", s Interneta, <http://infolab.stanford.edu/~wangz/project/imsearch/review/JOUR/datta.pdf>, srpanj 2022.

## *Bibliografija*

- [10] Latif, A.; Rasheed, A.; Sajid, U.; "Content-Based Image Retrieval and Feature Extraction: A Comprehensive Review", s Interneta, <https://www.hindawi.com/journals/mpe/2019/9658350/>, srpanj 2022.
- [11] Shubhankar Reddy, K.; Sreedhar, K.; "Image Retrieval Techniques: A Survey", s Interneta, [https://www.ripublication.com/irph/ijece16/ijecev9n1\\_04.pdf](https://www.ripublication.com/irph/ijece16/ijecev9n1_04.pdf), srpanj 2022.
- [12] Mohd Zin, N. A.; Yusof, R.; Lashari, S. A. i ostali; "Content-Based Image Retrieval in Medical Domain: A Review", s Interneta, <https://iopscience.iop.org/article/10.1088/1742-6596/1019/1/012044>, srpanj 2022.
- [13] Autoencoder, s Interneta, <https://deepai.org/machine-learning-glossary-and-terms/autoencoder>, srpanj 2022.
- [14] Autoencoder, s Interneta, <https://en.wikipedia.org/wiki/Autoencoder>, srpanj 2022.
- [15] Nearest Neighbors, s Interneta, <https://scikit-learn.org/stable/modules/neighbors.html>, srpanj 2022.

# Pojmovnik

**CBIR** pretraživanje slika po sadržaju (engl. content-based image retrieval). 2, 14, 15, 24

**CT** računalna tomografija (engl. computed tomography). 3, 9

**DICOM** Digital Imaging and Communications in Medicine. 1, 3–5, 9, 10, 12, 15, 41

**MRI** magnetska rezonancija (engl. magnetic resonance imaging). 3, 9

**NEMA** National Electrical Manufacturers Association. 4

**PACS** Picture archiving and communication system. 1, 3, 4, 8, 9, 15, 24, 39–41

**PDF** Portable Document Format. 3

**PF** rendgenski film (engl. X-ray plain film). 3

# Sažetak

PACS sustavi kliničko-bolničkih centara mogu sadržavati ogroman broj podataka. To otežava analizu i obradu njihovih podataka potrebnih za pretraživanje semantički sličnih slika dobivenim medicinskim snimanjima. U ovome radu opisane su tehnike pretraživanja takvih slika te je ponuđeno rješenje u obliku strojnog učenja, uz prikaz rezultata. Rezultati rješenja su zadovoljavajući, no postoji nekoliko načina za poboljšanje pretraživanja.

***Ključne riječi*** — CBIR, PACS, DICOM, pretraživanje slika, strojno učenje

## Abstract

PACS repositories of clinical centers and hospitals can contain a huge amount of data. This makes it difficult to analyze and process their data, which is necessary to search for semantically similar images obtained from medical imaging. In this paper, the methods for image retrieval of such images are described and a solution in the form of machine learning is offered, along with a display of the results. The results of the solution are satisfactory, but there are several ways to improve the search.

***Keywords*** — CBIR, PACS, DICOM, image retrieval, machine learning

# Dodatak A

## Naslov priloga

U digitalnoj verziji ovoga rada priložen je dodatak A koji sadrži programski kod prototipa rješenja pretraživanja slika PACS sustava po sadržaju.