

Univerzitet u Novom Sadu

Fakultet tehničkih nauka

Klasifikacija slika mamografskih pregleda

Student: Nikola Ljubičić IT 44/2021

Sadržaj

1. Uvod.....	1
2. Struktura projekta i opis fajlova.....	1
2.1 config.py	1
2.2 data_preparation.py.....	1
2.3 dataset.py	1
2.4 model.py.....	2
2.5 train.py	2
2.6 evaluate.py	3
2.7 inference.py.....	3
2.8 requirements.txt	5
3. Objašnjenje metrika	5
4. Zaključak.....	5

1. Uvod

Cilj ovog projekta bio je da se napravi model koji može da klasifikuje mamografske snimke dojki kao sumnjive (*maligne*) ili nesumnjive (*benigne*) promene.

Ideja je da model funkcioniše kao asistent radiologu i pomogne u ranom otkrivanju raka dojke.

Za izradu ovog projekta, korišćen je javno dostupan dataset sa Kaggle platforme:

<https://www.kaggle.com/datasets/awsaf49/cbis-ddsm-breast-cancer-image-dataset>

2. Struktura projekta i opis fajlova

2.1 config.py

Ovaj fajl olakšava podešavanje i menjanje parametara

Sadrži osnovna podešavanja projekta:

- DATA_DIR, RAW_DIR, PROCESSED_DIR predstavljaju putanje do foldera sa slikama i obrađenim podacima.
- BATCH_SIZE, LEARNING_RATE, NUM_EPOCHS predstavljaju osnovne parametre treniranja kao što su veličina paketa, brzina učenja i broj epoha,
- IMAGE_SIZE predstavlja dimenzije na koje se slike prilagođavaju (224x224),
- DEVICE određuje da li se koristi CPU ili GPU. U ovom slučaju se koristi CPU zbog ograničenih resursa računara (integrisana grafička kartica).
- CLASS_MAPPING mapiranje koje pretvara oznake ("benign", "malignant") u brojeve 0 i 1

2.2 data_preparation.py

Priprema CSV fajlove i slike za model.

- find_jpeg_for_dcm(dcm_path) pronalazi odgovarajuću .jpg sliku za svaki DICOM zapis (na osnovu ID broja u nazivu fajla).
- prepare_dataset(csv_path) učitava CSV sa informacijama o slučajevima, povezuje ga sa .jpg slikama, čisti prazne vrednosti i pravi novu kolonu label (0 ili 1). Na kraju se formira index.csv koji sadrži samo ispravne slike sa pripadajućim oznakama (benigno/maligno).

2.3 dataset.py

Definiše klasu MammoDataset koja priprema slike pre nego što se pošalju modelu. U konstruktoru __init__ uključuje augmentaciju kako bi se dataset učinio robusijim. Metode primenjene u augmentaciji su nasumično okretanje, rotaciju i promenu kontrasta kako bi se slike malo razlikovale i model učio bolje.

`__getitem__` učitava sliku sa diska, konvertuje u RGB format jer je original crno-bela, smanjuje veličinu na 224x224 piksela i vraća je zajedno sa oznakom label. Ova klasa omogućava da PyTorch lako učitava slike u serijama (batch-ovima).

2.4 model.py

Definiše neuronsku mrežu.

`create_model(pretrained=True, fine_tune_layers=3)` koristi MobileNetV3 Small unapred treniranu konvolucionu mrežu poznatu po brzini i maloj veličini. Poslednji sloj modela menja se tako da ima samo jedan izlaz potreban za binarnu klasifikaciju. Svi slojevi su zamrznuti osim poslednjih nekoliko, što znači da se treniraju samo zadnji nivo mreže, tj vrši se “fine-tuning”. Ovim pristupom model zadržava već naučene vizuelne osobine, a uči da prepozna konkretno maligne i benigne snimke.

2.5 train.py

Glavni fajl koji pokreće proces obuke modela.

- `create_balanced_sampler(df)` izjednačava broj primera iz obe klase da bi trening bio pravilno raspoređen.
- `train_epoch()` izvršava jednu epohu treniranja, prolazi kroz slike, računa grešku i ažurira težine modela.
- `validate()` proverava model posle svake epohe, izračunava AUROC i F1-score.
- `train()` povezuje sve, deli podatke na trening i validaciju, pravi model, pokreće više epoha, i čuva najbolji model. Na kraju prikazuje najbolji rezultat modela i gde je fajl sačuvan.

```
!python src/train.py

Training on device: cpu
Loading dataset...
Total images: 397
Benign: 191 | Malignant: 206
Creating model...
Starting training...

Epoch 1/8: Loss=0.7039, AUROC=0.596, F1=0.000
  Best model saved (AUROC: 0.596)
Epoch 2/8: Loss=0.6558, AUROC=0.594, F1=0.174
Epoch 3/8: Loss=0.6646, AUROC=0.590, F1=0.367
Epoch 4/8: Loss=0.6569, AUROC=0.593, F1=0.582
Epoch 5/8: Loss=0.6143, AUROC=0.551, F1=0.565
Epoch 6/8: Loss=0.5822, AUROC=0.620, F1=0.548
  Best model saved (AUROC: 0.620)
Epoch 7/8: Loss=0.5806, AUROC=0.612, F1=0.673
Epoch 8/8: Loss=0.5945, AUROC=0.598, F1=0.647
Training complete! Best AUROC: 0.620
Model saved to: models/best_model.pt
```

Slika 1. Train

2.6 evaluate.py

Procena performansi gotovog modela. Učitava index.csv deli podatke na test skup i pravi DataLoader. Učitava najbolji model i računa glavne metrike:

- **AUROC**
- **F1-score**,
- **Precision**,
- **Recall**,
- i prikazuje **matricu konfuzije**.

Na kraju crta i čuva ROC krivu, koja pokazuje koliko je model dobar u prepoznavanju pozitivnih slučajeva.

```
!python src/evaluate.py

Evaluating model...
Validation Metrics:
  AUROC:      0.6203
  F1 Score:   0.5479
  Precision:  0.6452
  Recall:     0.4762
Confusion Matrix:
  [[27 11]
   [22 20]]
ROC curve saved to: roc_curve.png
```

Slika 2. Evaluate

2.7 inference.py

Koristi se da za testiranje modela na novim slikama. Ovaj deo predstavlja kako bi sistem mogao da radi u praksi, lekar bi ubacio snimak, a model bi vratio procenu.

- `load_model()` učitava već treniran model.
- `preprocess_image()` priprema jednu sliku da bude istog formata kao u treningu (crno-bela, smanjena, normalizovana).
- `predict(img_path)` šalje sliku kroz model i vraća rezultat, tj. verovatnoću i da li je slika *benigna* ili *maligna*.

```
!python src/inference.py data/raw/jpeg/1.3.6.1.4.1.9590.100.1.2.129575250012593782010605863692289437698/1-105.jpg
```

Prediction for: data/raw/jpeg/1.3.6.1.4.1.9590.100.1.2.129575250012593782010605863692289437698/1-105.jpg
Class: MALIGNANT
Probability: 77.48%
Suspicious: YES

Slika 3. Rezultati predikcije

```
!python src/inference.py data/raw/jpeg/1.3.6.1.4.1.9590.100.1.2.206548514011121207219994124060615579686/1-171.jpg
```

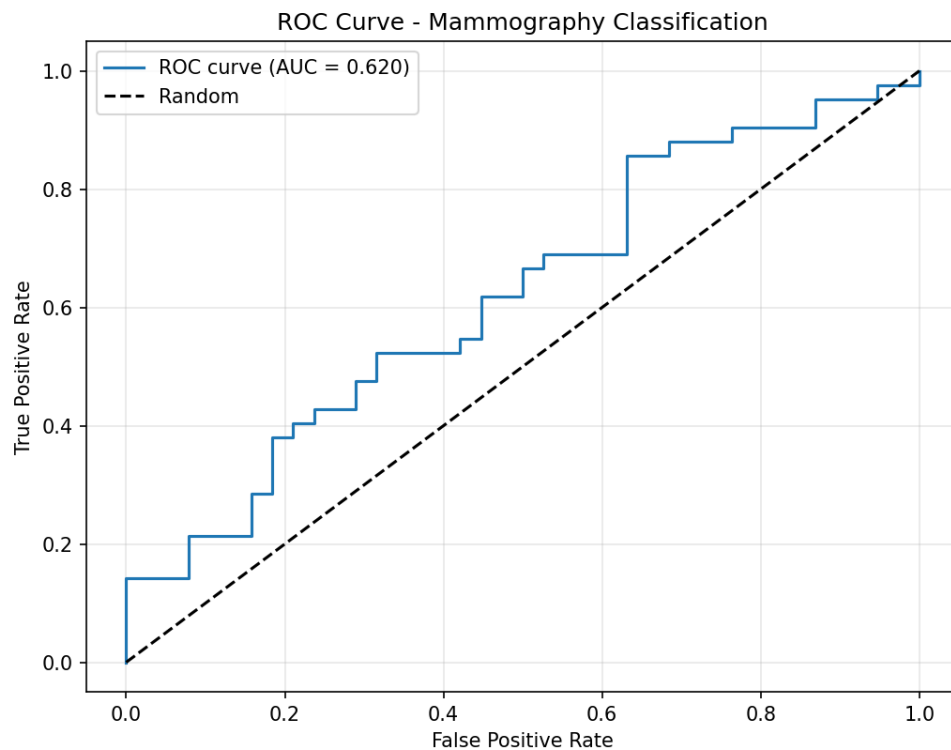
Prediction for: data/raw/jpeg/1.3.6.1.4.1.9590.100.1.2.206548514011121207219994124060615579686/1-171.jpg
Class: MALIGNANT
Probability: 81.19%
Suspicious: YES

Slika 4. Rezultati predikcije

```
!python src/inference.py data/raw/jpeg/1.3.6.1.4.1.9590.100.1.2.73898800613320269625592651560791507194/1-241.jpg
```

Prediction for: data/raw/jpeg/1.3.6.1.4.1.9590.100.1.2.73898800613320269625592651560791507194/1-241.jpg
Class: BENIGN
Probability: 45.84%
Suspicious: NO

Slika 5. Rezultati predikcije



Slika 6. Rezultati predikcije

2.8 requirements.txt

Navodi sve potrebne biblioteke, torch, torchvision, opencv-python, albumentations, scikit-learn, pandas, numpy, matplotlib. To omogućava da se projekat lako pokrene na drugom računaru.

3. Objašnjenje metrika

F1-score meri koliko model dobro balansira između preciznosti (Precision) i odziva (Recall). Koristi se kada su klase neizbalansirane. Što je vrednost bliža 1 znači da model dobro prepoznaje obe klase, dok niže vrednosti ukazuju da model pravi više grešaka u jednoj od klasa. U ovom projektu F1-score od oko 0.55 pokazuje da model ima solidan balans između tačnih pogodaka i promašaja, s obzirom da se zbog ograničenih resursa radilo na malom skupu podataka.

Matrica konfuzije

Rezultat evaluacije:

```
[[27 11]
```

```
[22 20]]
```

Ova matrica pokazuje:

- 27 slučajeva koji su stvarno benigni i tačno prepoznati (True Negatives).
- 20 slučajeva koji su stvarno maligni i tačno prepoznati (True Positives).
- 11 zdravih slika pogrešno označenih kao sumnjive (False Positives).
- 22 sumnjive slike koje su promašene i označene kao zdrave (False Negatives).

To znači da model bolje prepoznaje zdrave slučajeve, dok ponekad propušta sumnjive, što je uobičajeno kod manjih datasetova.

4. Zaključak

S obzirom da je korišćeno manje od 400 slika i da je trening rađen bez GPU-a, postignuti rezultati su zadovoljavajući. Model pokazuje potencijal i jasno razlikuje osnovne obrasce između benignih i malignih slika. U budućnosti bi se tačnost mogla poboljšati dodavanjem više podataka, korišćenjem GPU-a i dubljih modela. Projekat demonstrira čitav proces razvoja sistema za medicinsku klasifikaciju od pripreme podataka, preko obuke, do evaluacije i testiranja modela.