



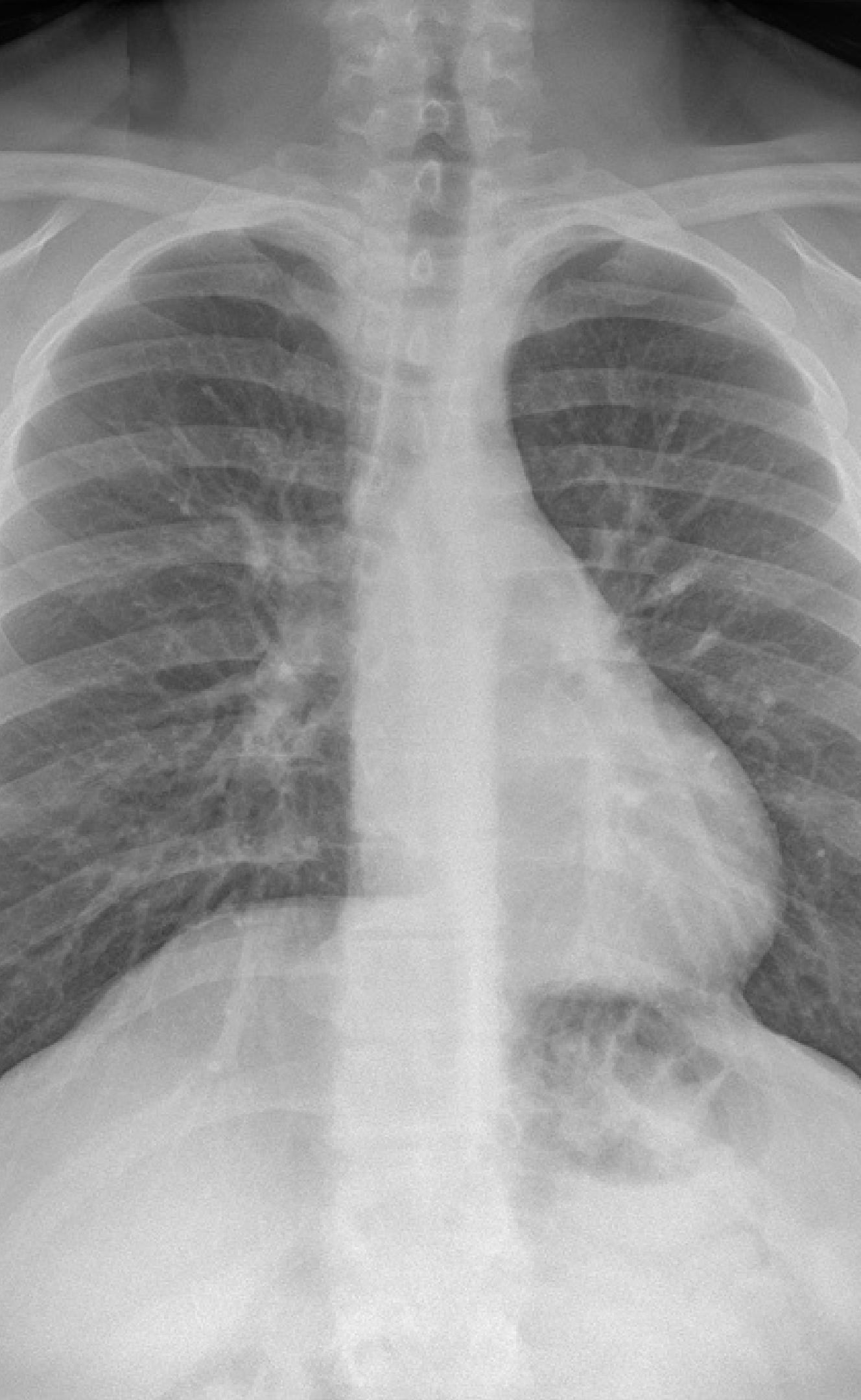
FAKULTET TEHNIČKIH NAUKA  
NOVI SAD 2023

# Prepoznavanje prisustva raka na rendgenskim snimcima pluća

---

Teodora Jezdimirović E1 99/2022  
Aleksandra Zarija E1 98/2022

Profesor: Vladimir Petrović  
Asistent: Nebojša Božanić



# Uvod

- Zadatak: klasifikacija rendgenskih slika pluća u 2 klase u zavisnosti od toga da li na njima postoji rak ili ne pomoću neuralne mreže
- Baza sadrži 6652 x-ray slike, 6652 .xml fajla koji sadrže informacije o slikama i 1037 .txt fajlova koji sadrže koordinate značajnih tačaka na plućima

# Njihovo rešenje

---

# *Multi-Label Chest X-Ray Classification via Deep Learning*

- Klasifikacija 224 316 x-ray slika u 14 klasa
- Upoređeni su rezultati dobijeni sa 5 različitih modela
- Trening skup sadrži 80% podataka, a test skup sadrži preostalih 20%

# Hiperparametri sa kojima se postižu najbolji rezulati

- Veličina *batch-a*: 96
- Funkcija cene: Binarna unakrsna entropija
- Optimizacioni algoritam: Adam
- Brzina učenja: 0,001 – 0,00001
- Broj epoha: 40
- Broj parametara modela: ~497k
- Postignuta tačnost: 85%

# Model CustomNet

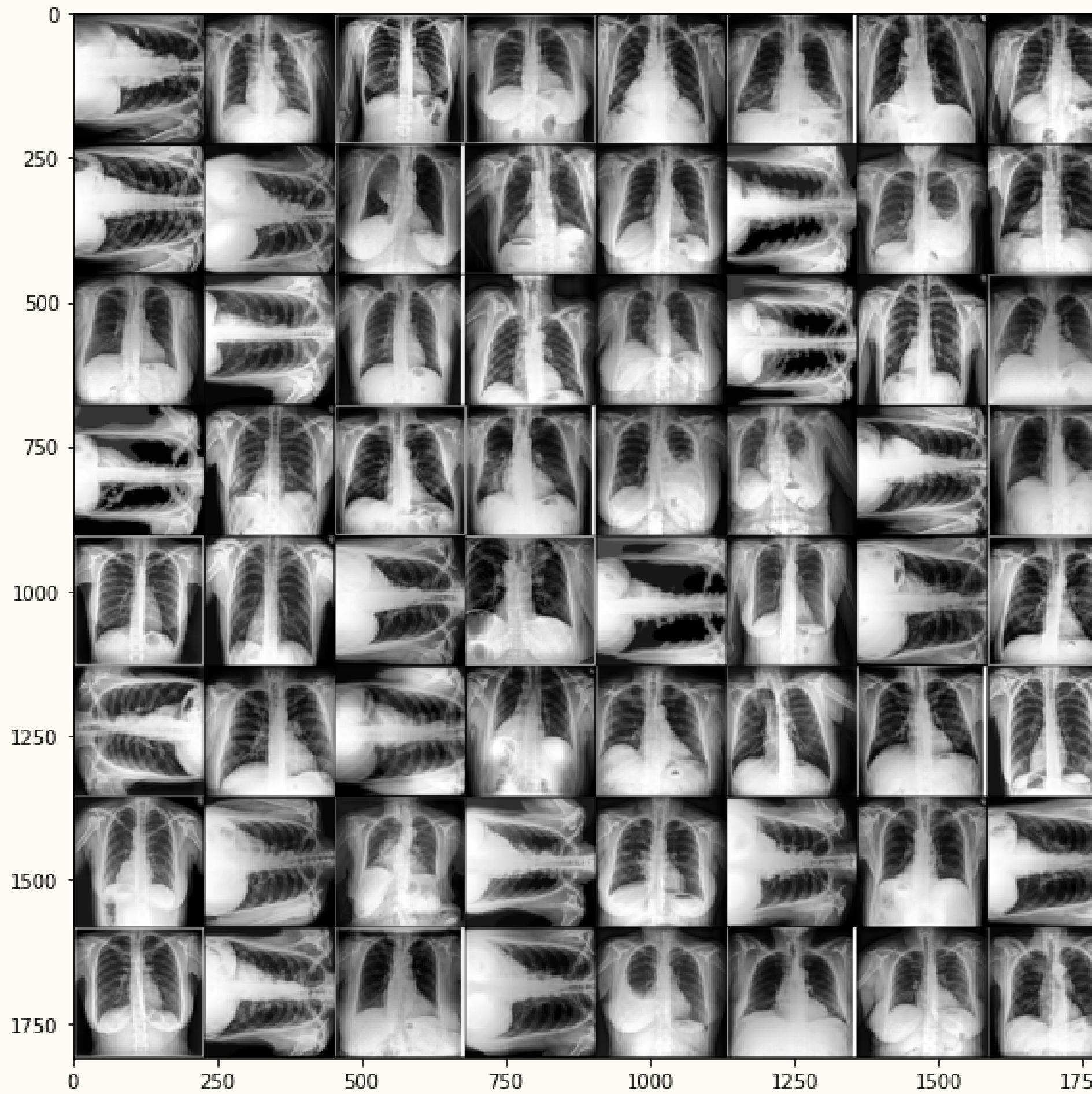
```
#model
class CustomNet(nn.Module):
    def __init__(self, num_classes=2, is_trained=False):
        super().__init__()
        self.ConvLayer1 = nn.Sequential(
            nn.Conv2d(3, 8, 3), # inp (3, 512, 512)
            nn.Conv2d(8, 16, 3),
            nn.BatchNorm2d(16), #
            nn.MaxPool2d(2),
            nn.ReLU() # op (16, 256, 256)
        )
        self.ConvLayer2 = nn.Sequential(
            nn.Conv2d(16, 32, 5), # inp (16, 256, 256)
            nn.Conv2d(32, 32, 3),
            nn.BatchNorm2d(32), #
            nn.MaxPool2d(4),
            nn.ReLU() # op (32, 64, 64)
        )
        self.ConvLayer3 = nn.Sequential(
            nn.Conv2d(32, 64, 3), # inp (32, 64, 64)
            nn.Conv2d(64, 64, 5),
            nn.BatchNorm2d(64), #
            nn.MaxPool2d(2),
            nn.ReLU() # op (64, 32, 32)
        )
        self.ConvLayer4 = nn.Sequential(
            nn.Conv2d(64, 128, 5), # inp (64, 32, 32)
            nn.Conv2d(128, 128, 3),
            nn.BatchNorm2d(128), #
            nn.MaxPool2d(2),
            nn.ReLU() # op (128, 16, 16)
        )
        #self.Lin1 = nn.Linear(15488, 15)
        self.Lin1 = nn.Sequential(nn.Linear(512, 1), nn.Sigmoid())
```

# Naše rešenje

---

# Naše rešenje

- Osnovni *data set* sadrži 6652 x-ray slike, od kojih je samo 3851 slika korisna (705 sa rakom i 3146 bez raka)
- Postoji opcija za dinamičko određivanje dimenzija slika, ali nije inkorporirana u finalno rešenje
- Trening skup sadrži 75% podataka, validacioni 15%, a test skup sadrži preostalih 10%.

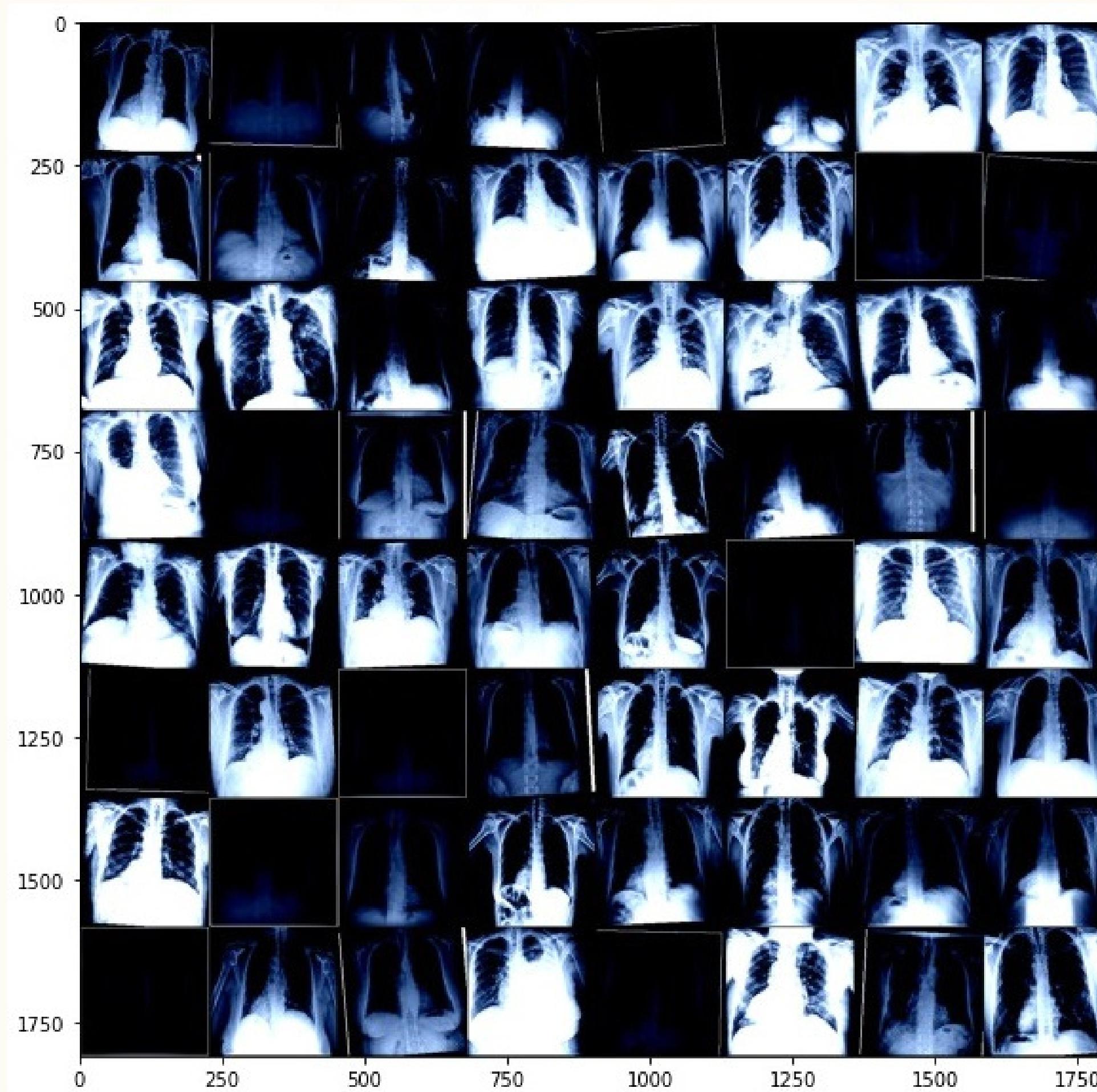


Prikaz  
batch-a

# Naše rešenje

- Sve slike su normalizovane, a nad slikama u trening skupu su urađene i translacija, skaliranje i rotacija.
- Slika na kojima postoji rak ima 705, a onih na kojima nema raka 3146 zbog čega je urađeno balansiranje *batch-a*

Prikaz  
*batch-a*  
na trening  
skupu



# Hiperparametri sa kojima se postižu najbolji rezulati

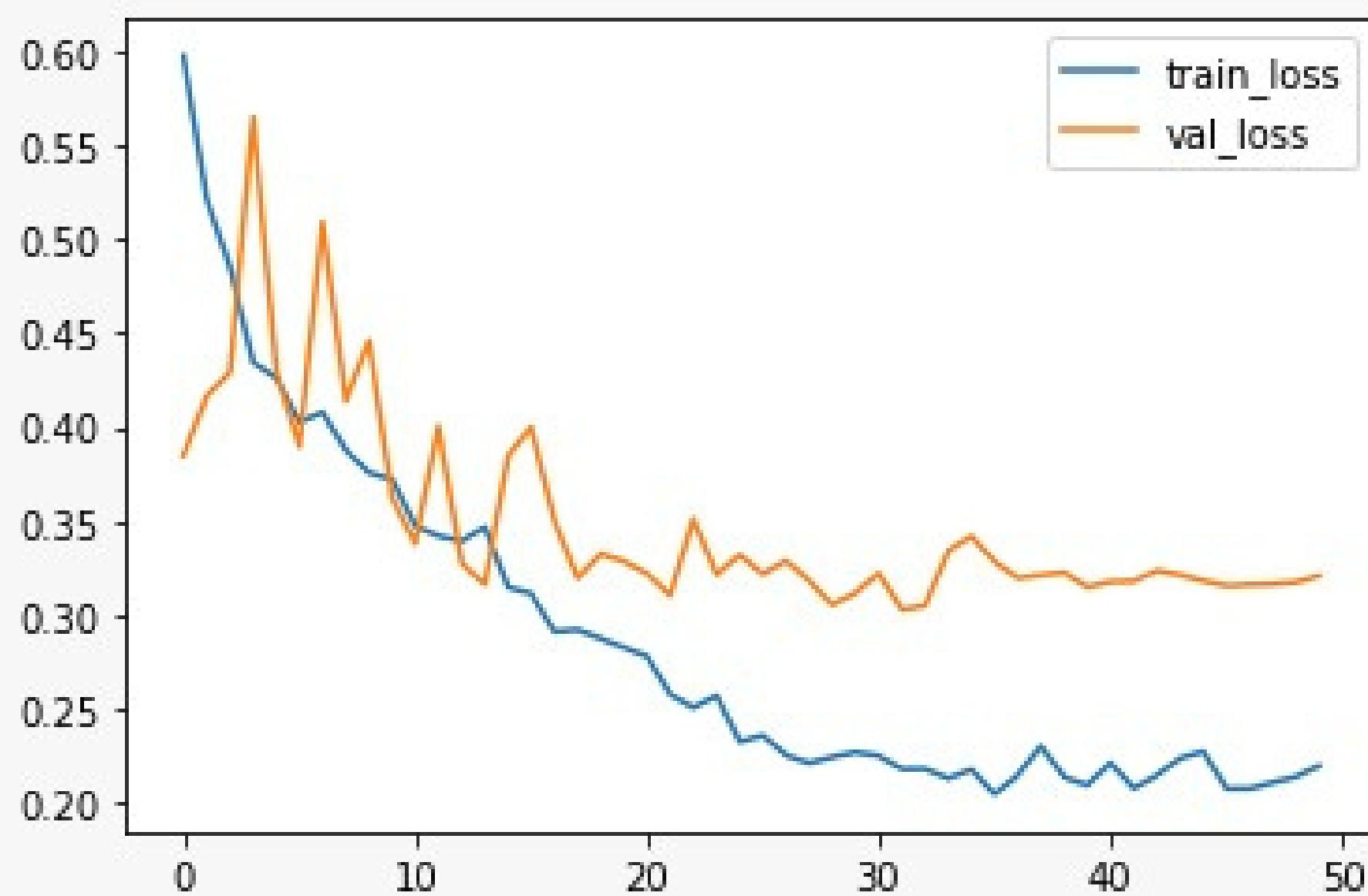
- Veličina *batch-a*: 64 na trening skupu i 16 na validacionom i test skupu
- Funkcija cene: Binarna unakrsna entropija
- Optimizacioni algoritam: AdamW
- Brzina učenja: 0,001 - 0,00001
- Broj epoha: 50
- Broj parametara modela: ~26k
- Regularizacija: L2 sa  $\lambda = 0,0001$
- Postignuta tačnost: 87,5%, 82,4% za klasu postoji rak i 88,4% za klasu ne postoji rak

# Modifikovani model CustomNet

```
Model
class CustomNet(nn.Module):
    def __init__(self, num_classes=2, is_trained=False):
        super().__init__()
        self.ConvLayer1 = nn.Sequential(
            nn.Conv2d(3, 8, 3), # inp (3, 512, 512)
            nn.Conv2d(8, 16, 3),
            nn.BatchNorm2d(16), #
            nn.MaxPool2d(2),
            nn.ReLU() # op (16, 256, 256)
        )
        self.ConvLayer2 = nn.Sequential(
            nn.Conv2d(16, 32, 5), # inp (16, 256, 256)
            nn.Conv2d(32, 32, 3),
            nn.BatchNorm2d(32), #
            nn.MaxPool2d(4),
            nn.ReLU() # op (32, 64, 64)
        )
        self.ConvLayer3 = nn.Sequential(
            nn.Conv2d(32, 64, 3), # inp (32, 64, 64)
            nn.Conv2d(64, 128, 5), #pazi ovde je bilo 64!
            nn.BatchNorm2d(128), #pazi ovde je bilo 64!
            nn.MaxPool2d(2),
            nn.ReLU() # op (64, 32, 32)
        )
        # self.ConvLayer4 = nn.Sequential(
        #     nn.Conv2d(64, 128, 5), # inp (64, 32, 32)
        #     nn.Conv2d(128, 128, 3),
        #     nn.BatchNorm2d(128), #
        #     nn.MaxPool2d(2),
        #     nn.ReLU() # op (128, 16, 16)
        # )
        #self.Lin1 = nn.Linear(15488, 15)
        self.Lin1 = nn.Sequential(nn.Linear(12800, 1), nn.Sigmoid())
    
```

# Rezultati

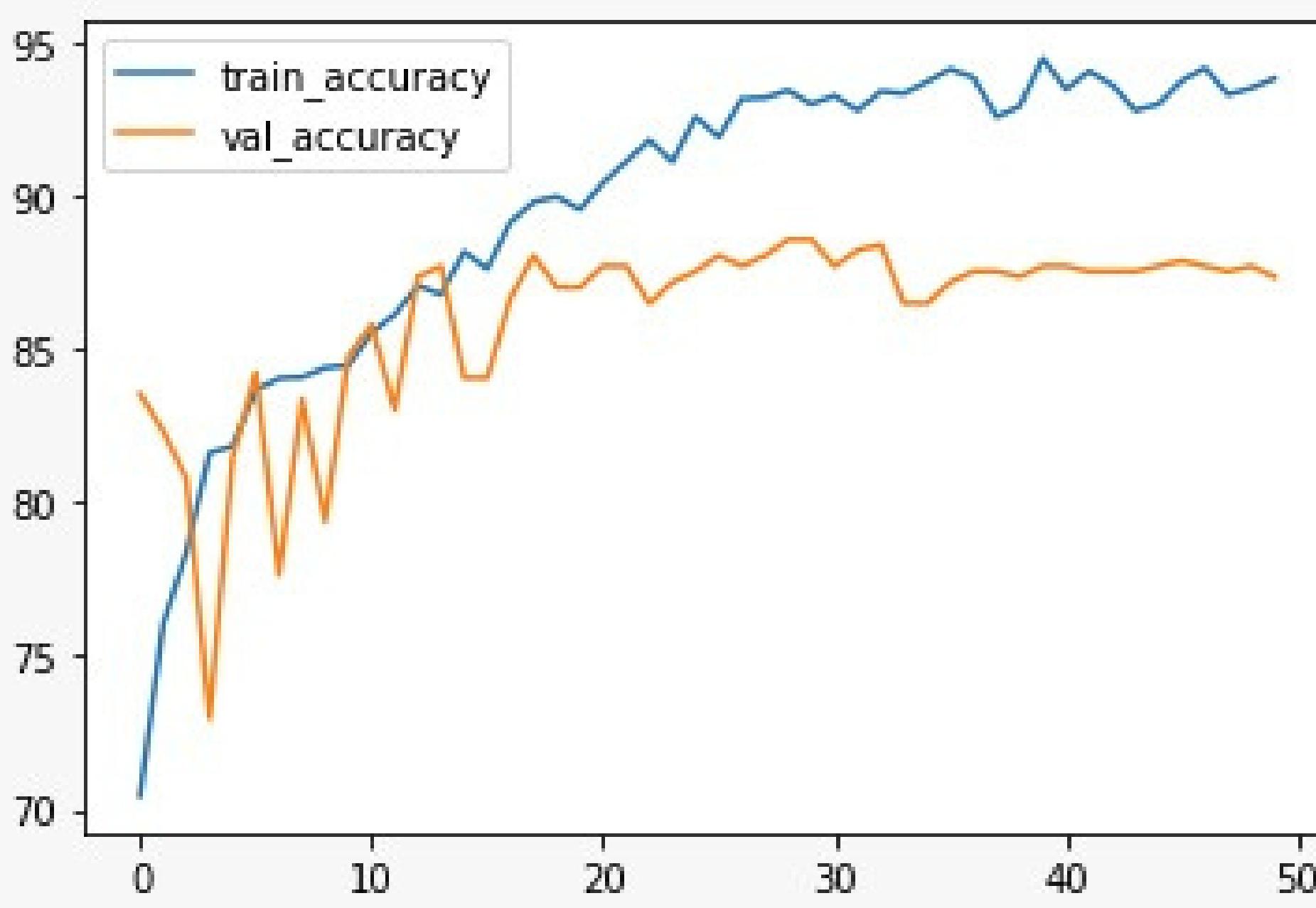
---



# Funkcija cene

## *Binary Cross Entropy -BCE*

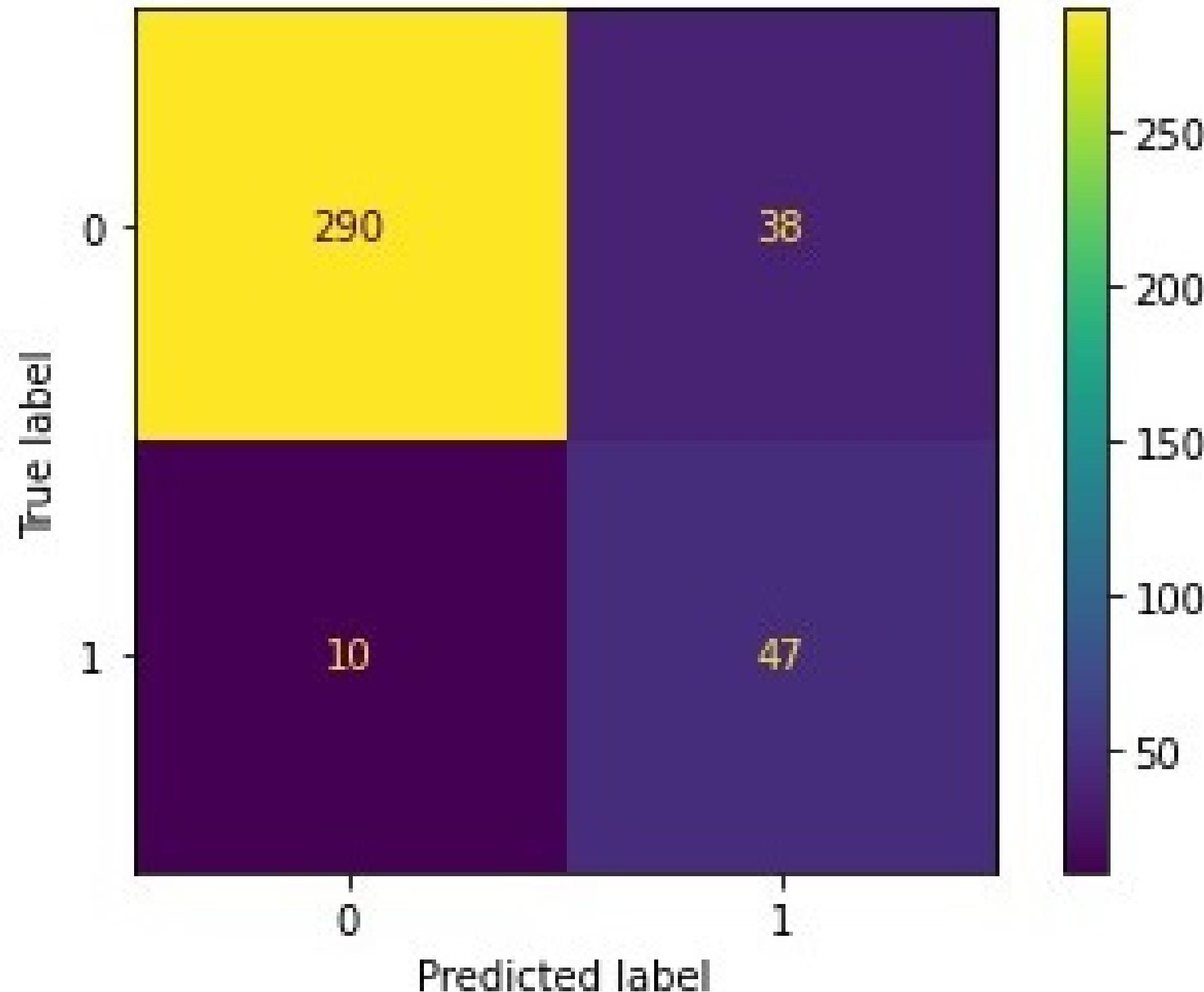
- Cilj je minimizacija *loss-a*
- Krive se razilaze od 20. epohe



# Tačnost

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

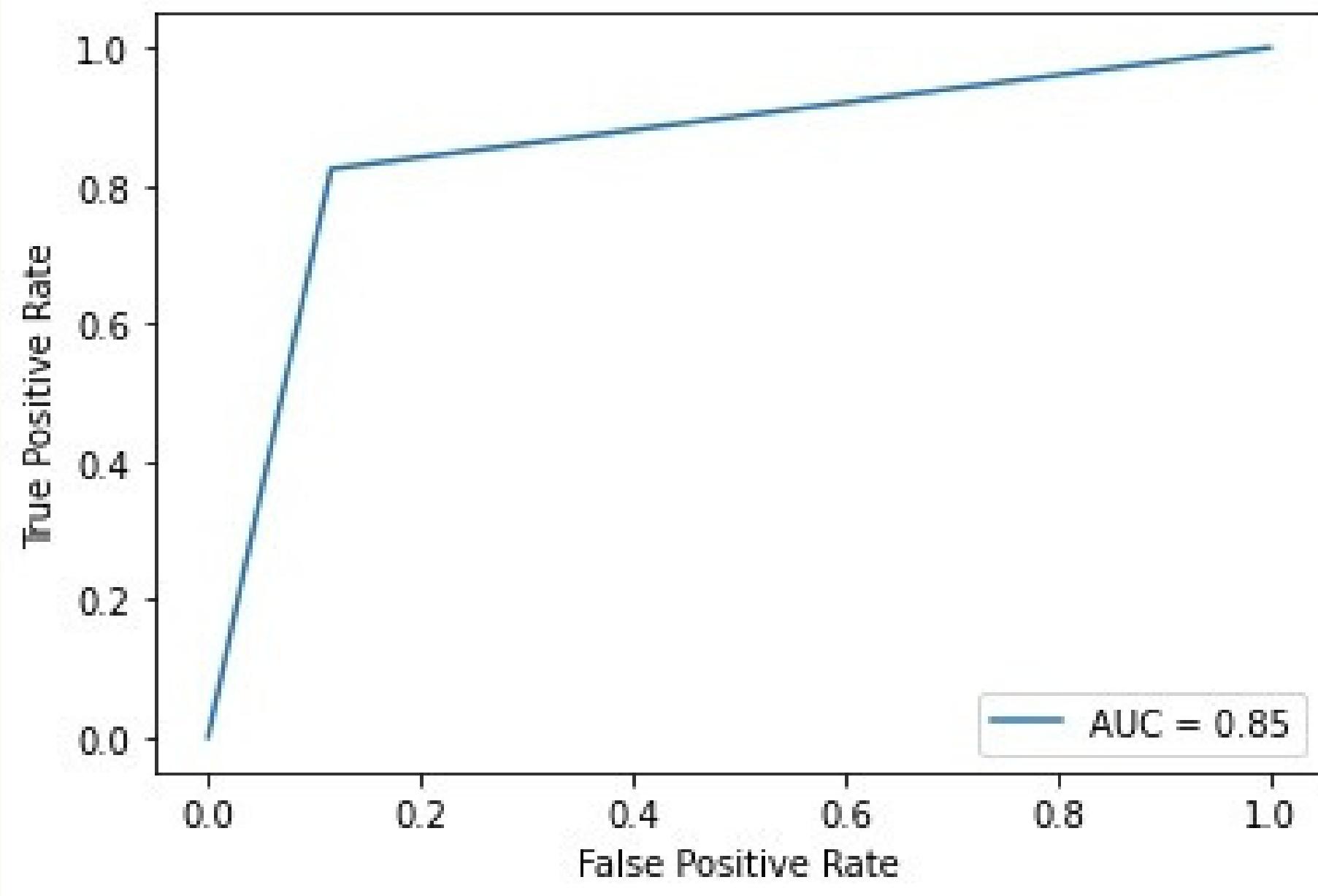
- Krive se razilaze od 20. epohe



# Matrica konfuzije

*True positive: 47*  
*False positive: 38*

*True negative: 290*  
*False negative: 10*



# ROC kriva

Grafički prikaz mogućnosti binarnog klasifikatora da razdvaja klase.

- *Area under the ROC curve* – AUC – Površina ispod krive (AUC) je mera sposobnosti binarnog klasifikatora da pravi razliku između klasa i koristi se kao rezime ROC krive.

# Reference

---

Pillai, A.S. (2022)  
*Multi-Label Chest X-Ray Classification via  
Deep Learning. Journal of Intelligent Learning Systems and Applications, 14, 43–56.*  
<https://doi.org/10.4236/jilsa.2022.144004>

**Hvala na pažnji!**

---