



UNIVERZITET U ZENICI

Politehnički fakultet

Softversko inženjerstvo

Rudarenje podataka

## KLASIFIKACIJA SLIKA PASA I MAČAKA

*Projekat*

Studenti:

Alena Jonjić, II–117

Lamija Čolaković, II–118

Profesor:

doc. dr. Adnan Dželihodžić

Asistenti:

Faris Hambo, Ahmed Mujić

Zenica, 2025.

## Sadržaj

1. OBJAŠNJENJE KODA	3
BIBLIOTEKE	3
PARAMETRI	3
PRETPROCESIRANJE I SPLIT	4
CNN MODEL	5
TRENIRANJE MODELA	6
SPREMANJE MODELA	7
EVALUACIJA: PRECIZNOST, RECALL, F1 SCORE	7
MATRICA KONFUZIJE	8
PRIKAZ I PREDIKCIJA JEDNE SLIKE IZ TEST SKUPA ZA SVAKU KLASU	9
FUNKCIJA ZA PRIKAZ SLIKE I PREDIKCIJE	9
PRIKAZ JEDNE MAČKE I JEDNOG PSA	10
FUNKCIJA ZA PREDIKCIJU PO NAZIVU FILE-A	11
2. OPIS PROJEKTA	12
3. OPIS DATASET-A	12
4. OPIS PROBLEMA	12
5. OPIS ALGORITMA	13
6. METODE KORIŠTENE ZA POBOLJŠANJE ALGORITMA	13
7. REZULTATI I NJIHOVA INTERPRETACIJA	13

# 1. OBJAŠNJENJE KODA

## BIBLIOTEKE

```
import random

import numpy as np

import tensorflow as tf

from tensorflow.keras import layers, models, optimizers

from tensorflow.keras.preprocessing import image

import matplotlib.pyplot as plt

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from sklearn.metrics import classification_report, confusion_matrix

import seaborn as sns

import os
```

- random, numpy: Za rad s brojevima i nizovima.
- tensorflow, keras: Glavna biblioteka za gradnju neuronskih mreža.
- matplotlib.pyplot: Za crtanje grafova (loss, tačnost, prikaz slika).
- ImageDataGenerator: Automatsko učitavanje slika i augmentacija.
- classification\_report, confusion\_matrix: Za mjerenje performansi modela.
- seaborn: Za crtanje matrice konfuzije.

## PARAMETRI

```
IMG_SIZE = (100, 100)

BATCH_SIZE = 64

EPOCHS = 15
```

IMG\_SIZE: Sve slike će biti smanjene ili povećane na 100x100 piksela.

BATCH\_SIZE: Model će trenirati s 64 slike istovremeno.

EPOCHS: Model će 15 puta proći kroz sve trening slike.

## PRETPROCESIRANJE I SPLIT

```
datagen = ImageDataGenerator(  
    rescale=1./255,  
    validation_split=0.2  
)
```

Slike se normalizuju (vrijednosti piksela 0–255 postaju 0–1).

Podaci se dijele na 80% trening i 20% validacija.

```
train_generator = datagen.flow_from_directory(  
    directory='.', # ovdje je folder sa podfolderima Cat/ i Dog/  
    target_size=IMG_SIZE,  
    batch_size=BATCH_SIZE,  
    class_mode='binary',  
    subset='training',  
    shuffle=True  
)
```

Pretražuje sve podfoldere (Cat/, Dog/).

Slike se učitavaju i randomizuju za trening.

```
validation_generator = datagen.flow_from_directory(  
    directory='.',  
    target_size=IMG_SIZE,  
    batch_size=BATCH_SIZE,  
    class_mode='binary',  
    subset='validation',  
    shuffle=False  
)  
  
print("Klase:", train_generator.class_indices)
```

Za validaciju se koristi isti set slika, ali redoslijed ostaje isti (zbog evaluacije).

## CNN MODEL

```
model = models.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', padding='same',
input_shape=(100, 100, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(32, (3, 3), activation='relu', padding='same'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(1, activation='sigmoid')
])
```

Conv2D: Prvi sloj ekstrakcije osobina (32 filtera 3x3, padding zadržava dimenziju).

MaxPooling2D: Smanjuje dimenziju slike uzimanjem maksimuma (2x2 blokovi).

Drugi Conv2D i MaxPooling2D: Još jedno izvlačenje značajki.

Flatten: Transformiše 2D mapu u 1D niz.

Dense(64): Sloj sa 64 neurona i ReLU aktivacijom.

Dense(1) + sigmoid: Izlazna jedinica za binarnu klasifikaciju (0 = Cat, 1 = Dog).

```
model.compile(optimizer=optimizers.Adam(learning_rate=0.001),
               loss='binary_crossentropy',
               metrics=['accuracy'])
```

Adam: Popularni optimizator koji automatski prilagođava learning rate.

binary\_crossentropy: Gubitak (loss) za binarnu klasifikaciju.

## TRENIRANJE MODELA

```
history = model.fit(  
    train_generator,  
    epochs=EPOCHS,  
    validation_data=validation_generator  
)
```

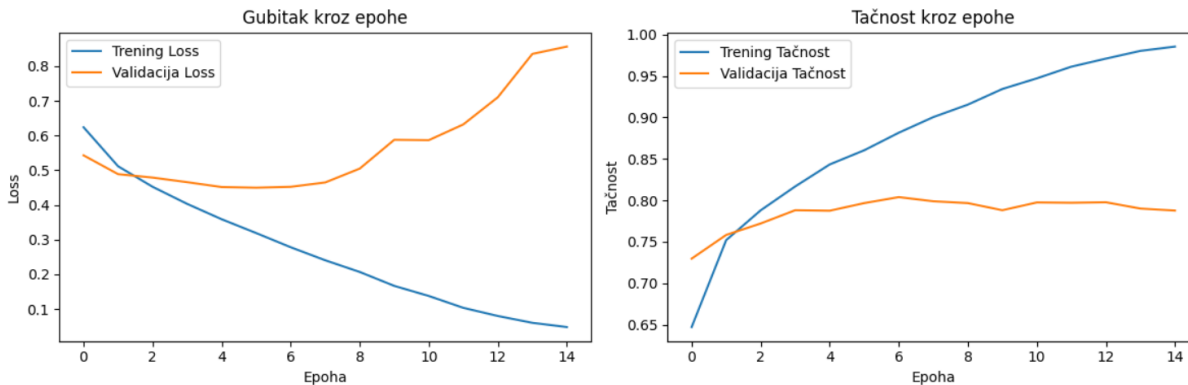
Pokreće proces treniranja modela kroz 15 epoha.

```
plt.figure(figsize=(12, 4))  
plt.subplot(1, 2, 1)  
plt.plot(history.history['loss'], label='Trening Loss')  
plt.plot(history.history['val_loss'], label='Validacija Loss')  
plt.title('Gubitak kroz epohe')  
plt.xlabel('Epoha')  
plt.ylabel('Loss')  
plt.legend()
```

Prikazuje krivulje gubitka (loss) kroz epohe

```
plt.subplot(1, 2, 2)  
plt.plot(history.history['accuracy'], label='Trening Tačnost')  
plt.plot(history.history['val_accuracy'], label='Validacija Tačnost')  
plt.title('Tačnost kroz epohe')  
plt.xlabel('Epoha')  
plt.ylabel('Tačnost')  
plt.legend()  
plt.tight_layout()  
plt.show()
```

Prikazuje tačnost za trening i validaciju.



## SPREMANJE MODELA

```
model.save('cat_dog_classifier_model.h5')
print("Model spašen kao 'cat_dog_classifier_model.h5'")
```

Sprema se trenirani model u .h5 fajl za kasnije korištenje.

## EVALUACIJA: PRECIZNOST, RECALL, F1 SCORE

```
validation_generator.reset()
y_true = validation_generator.classes
y_pred_probs = model.predict(validation_generator)
y_pred = (y_pred_probs > 0.5).astype(int).reshape(-1)
print("\nClassification Report:")
print(classification_report(y_true, y_pred, target_names=['Cat', 'Dog']))
```

```
Classification Report:
              precision    recall  f1-score   support

    Cat         0.81         0.76      0.78         2499
    Dog         0.77         0.82      0.79         2499

 accuracy              0.79         4998
 macro avg           0.79         0.79      0.79         4998
weighted avg           0.79         0.79      0.79         4998
```

y\_true: Stvarne klase.

y\_pred\_probs: Vjerovatnoće da je slika pas.

y\_pred: Finalna klasifikacija (> 0.5 → Dog, <= 0.5 → Cat).

## MATRICA KONFUZIJE

```
print("\nClassification Report:")

print(classification_report(y_true, y_pred, target_names=['Cat', 'Dog']))

cm = confusion_matrix(y_true, y_pred)

plt.figure(figsize=(6, 4))

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Cat', 'Dog'],
            yticklabels=['Cat', 'Dog'])

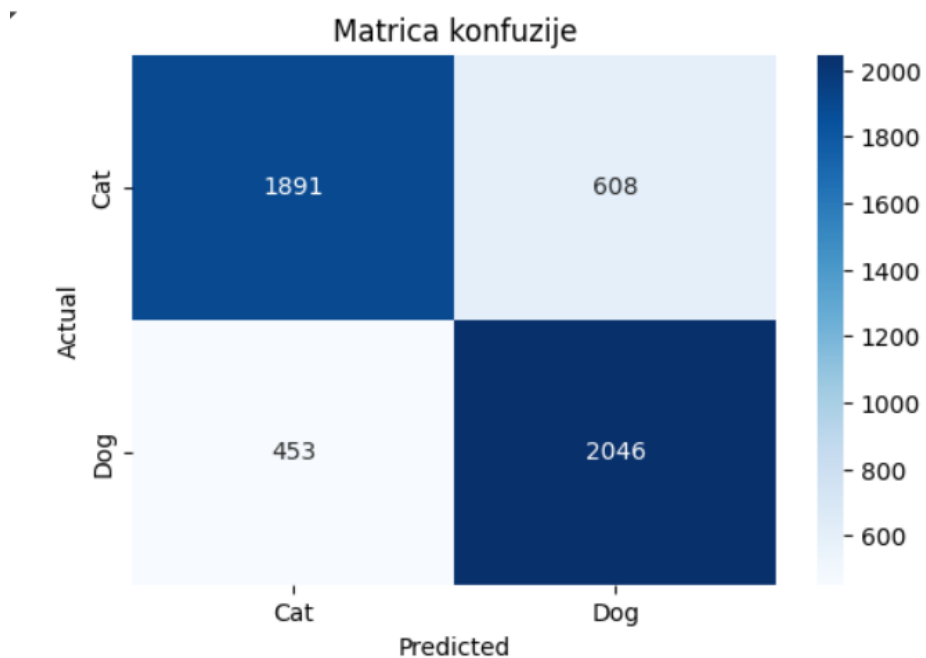
plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('Matrica konfuzije')

plt.show()
```

Prikazuje se izvještaj (preciznost, odziv, F1 score) i vizualna matrica konfuzije.





## PRIKAZ I PREDIKCIJA JEDNE SLIKE IZ TEST SKUPA ZA SVAKU KLASU

```
x_val, y_val = next(validation_generator)

for i in range(len(validation_generator)):

    batch_x, batch_y = next(validation_generator)

    x_val = np.concatenate((x_val, batch_x), axis=0)

    y_val = np.concatenate((y_val, batch_y), axis=0)
```

Kombinuju se svi validacijski batch-evi u jedinstvene x\_val i y\_val.

## FUNKCIJA ZA PRIKAZ SLIKE I PREDIKCIJE

```
def predict_and_show(image, true_label):

    pred_prob = model.predict(np.expand_dims(image, axis=0))[0][0]

    predicted_label = 'Dog' if pred_prob > 0.5 else 'Cat'

    true_label_str = 'Dog' if true_label == 1 else 'Cat'

    plt.imshow(image)

    plt.axis('off')

    plt.title(f'Stvarna klasa: {true_label_str}\nPredikcija: {predicted_label} ({pred_prob:.2f})')

    plt.show()
```

Funkcija prikazuje sliku i ispisuje:

- Stvarnu klasu (Cat/Dog)
- Predikciju i vjerovatnoću.

## PRIKAZ JEDNE MAČKE I JEDNOG PSA

```
for i in range(len(y_val)):
    if y_val[i] == 0: # Cat
        print("Prikaz slike mačke:")
        predict_and_show(x_val[i], y_val[i])
        break
for i in range(len(y_val)):
    if y_val[i] == 1: # Dog
        print("Prikaz slike psa:")
        predict_and_show(x_val[i], y_val[i])
        break
```

Prikazuje prvu sliku mačke i psa iz validacionog skupa.

Prikaz slike mačke:  
1/1 — 0s 292ms/step

Stvarna klasa: Cat  
Predikcija: Cat (0.01)



Prikaz slike psa:  
1/1 — 0s 65ms/step

Stvarna klasa: Dog  
Predikcija: Dog (1.00)



## FUNKCIJA ZA PREDIKCIJU PO NAZIVU FILE-A

```
def predict_from_filename(model, img_path, target_size=(100, 100),
class_names=['Cat', 'Dog']):

    # Učitavanje i priprema slike

    img = image.load_img(img_path, target_size=target_size)

    img_array = image.img_to_array(img) / 255.0

    img_array = np.expand_dims(img_array, axis=0)

    # Predikcija

    pred = model.predict(img_array, verbose=0)[0][0]

    pred_label = class_names[1] if pred > 0.5 else class_names[0]

    # Prikaz slike

    plt.imshow(img)

    plt.axis('off')

    plt.title(f'Predikcija: {pred_label} ({pred:.2f})')

    plt.show()

    return pred_label, float(pred)
```

Funkcija učitava sliku sa puta `img_path`, formatira je za model i prikazuje predikciju.

```
predict_from_filename(model, './Dog/245.jpg')
predict_from_filename(model, './Cat/110.jpg')
```

Primjeri testiranja slike direktno iz foldera (ručno testiranje).

Predikcija: Dog (0.99)



Predikcija: Cat (0.00)



## 2. OPIS PROJEKTA

Cilj ovog projekta je izgradnja modela duboke neuronske mreže zasnovane na konvolucijskim slojevima (CNN) za automatsku klasifikaciju slika pasa i mačaka. Model prepoznaje kojoj kategoriji (pas ili mačka) pripada data slika, koristeći skup podataka sa slikama koje su podijeljene u dva foldera: jedan sadrži slike mačaka, a drugi slike pasa. Projektom se nastoji postići visoka tačnost i pouzdanost klasifikacije, što može imati praktične primjene u oblasti računarskog vida i automatizacije.

## 3. OPIS DATASET-A

Dataset se sastoji od dva osnovna direktorija:

- Cat/ - sadrži slike mačaka
- Dog/ - sadrži slike pasa

Slike su različitih dimenzija i rezolucija, te su sve unaprijed procesirane i skalirane na veličinu 100x100 piksela radi konzistentnosti i optimizacije treninga modela. Dataset je podijeljen na trening set (80% slika) i validacijski set (20% slika) koristeći ImageDataGenerator iz TensorFlow Keras biblioteke sa automatskom augmentacijom i normalizacijom.

## 4. OPIS PROBLEMA

Klasifikacija slika je problem u oblasti računarskog vida gdje se želi automatski odrediti kategorija kojoj slika pripada. Pas i mačka su često korištene klase u takvim problemima zbog velike varijabilnosti u pozama, pozadinama i vrstama pasa i mačaka. Cilj je trenirati model da generalizuje i tačno prepozna slike neovisno o varijacijama. Problem se tretira kao binarna klasifikacija sa dvije klase: Cat i Dog.

## 5. OPIS ALGORITMA

Za rješavanje problema korišten je model konvolucijske neuronske mreže (CNN), koji se sastoji od dva konvolucijska sloja i dva sloja maksimalnog pooling-a za izvlačenje važnih vizuelnih osobina sa slika. Nakon njih slijedi potpuno povezani (dense) sloj sa 64 neurona i ReLU aktivacijom, te izlazni sloj sa jednim neuronom i sigmoid aktivacijom za binarnu klasifikaciju. Model se trenira koristeći Adam optimizator i funkciju gubitka `binary_crossentropy`. Slike su unaprijed obrađene kako bi se pikseli normalizovali na raspon  $[0,1]$ .

## 6. METODE KORIŠTENE ZA POBOLJŠANJE ALGORITMA

Normalizacija slika: Skaliranje vrijednosti piksela radi bržeg i stabilnijeg treniranja.

Podjela podataka sa validacijom: 20% podataka se koristi za validaciju radi praćenja overfittinga i optimizacije epoha.

Augmentacija podataka (ukoliko je korištena, npr. rotacije, translacije, skaliranja) može povećati generalizaciju modela (nije posebno navedeno u kodu, ali može se dodati).

Korištenje Adam optimizatora: adaptivni optimizator koji ubrzava konvergenciju.

Praćenje metrike tačnosti i gubitka tokom treninga omogućava pravovremeno zaustavljanje i bolju evaluaciju modela.

## 7. REZULTATI I NJIHOVA INTERPRETACIJA

Model je nakon 15 epoha treninga postigao zadovoljavajuću tačnost kako na trening, tako i na validacijskom skupu, što je prikazano kroz grafove gubitka i tačnosti. Izvještaj klasifikacije pokazuje preciznost, odziv i F1-score za obje klase (Cat i Dog), dok matrica konfuzije vizualno potvrđuje koliko su predikcije tačne. Model je sposoban tačno razlikovati slike pasa i mačaka sa visokom pouzdanošću. Takođe, prikazani su primjeri pojedinačnih slika iz validacijskog skupa sa njihovim stvarnim i predikcijskim oznakama, što dodatno potvrđuje uspješnost modela.