

# Dogs vs Cats Analyzer

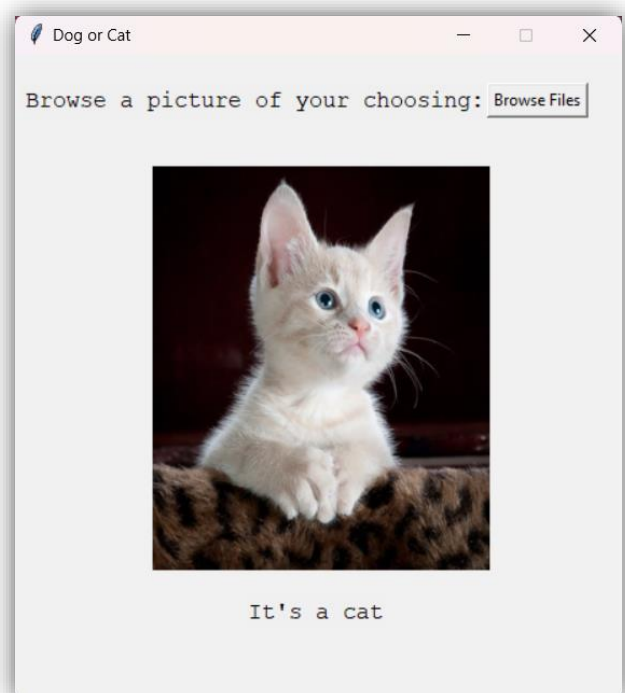
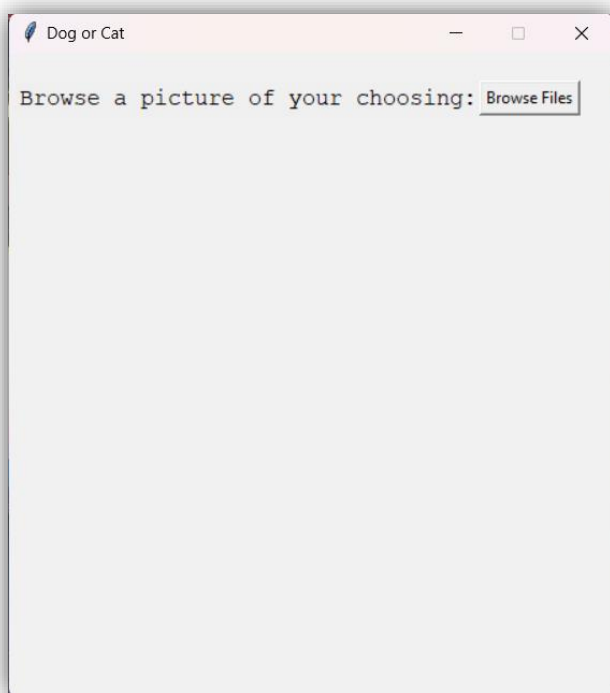
## Documentație pentru proiectul de la Inteligență Artificială

Mitroi Luca  
Informatică Aplicată  
Subgrupa 4

## Introducere

Programul realizat de mine are ca funcționalitate principală diferențierea între două tipuri de imagini. Imagini cu câini și imagini cu pisici. În momentul în care utilizatorul deschide aplicația, îi este cerut să introducă o imagine. După ce acesta a selectat o imagine cu care dorește să lucreze, aceasta va fi afișată pe ecran împreună cu textul “It’s a dog” sau “It’s a cat” în funcție de rezultatul pe care aplicația îl va da.

Mai jos se poate observa o imagine cu interfața aplicației în momentul deschiderii acesteia și după primirea răspunsului.



## Tehnologiile folosite

Pentru clasificarea imaginilor este utilizat un algoritm de deep learning. Acesta a fost antrenat utilizând un set de date de pe platforma Kaggle care conținea un număr de 25000 de imagini cu câini și pisici. Pentru realizarea acestui model am utilizat Google Colab, deoarece această platformă oferă o performanță mult mai ridicată decât orice calculator pe care îl am acasă, astfel s-a putut realiza antrenarea modelului într-un timp rezonabil.

Pentru realizarea interfeței grafice am utilizat librăria Tkinter a limbajului Python. Interfața grafică este un simplistă, ce conține doar un frame pe care se pun toate elemente aplicației, un text, un buton de Browse care accesează fișierele sistemului, un label pe care se inserează imaginea selectată și un text care informează utilizatorul în legătură cu decizia pe care a luat-o programul.

După ce a fost antrenat un model cu ajutorul platformei Google Colab acesta a fost exportat într-un fișier cu extensia .h5. Acesta a fost ulterior încărcat (loaded) în programul care conține interfața grafică. Acest lucru permite programului să funcționeze într-un timp rezonabil, nefiind nevoie să fie antrenat modelul.

În antrenarea modelului au fost utilizate 10 epoci (Epochs). Procesul a durat în medie 20 – 25 de minute, acest timp fiind posibil doar datorită echipamentelor performante oferite de către cei de la Google.

```
Epoch 1/10
49/49 [=====] - 167s 3s/step - loss: 0.8129 - accuracy: 0.5362 - val_loss: 0.6689 - val_accuracy: 0.5582
Epoch 2/10
49/49 [=====] - 158s 3s/step - loss: 0.6331 - accuracy: 0.6391 - val_loss: 0.5956 - val_accuracy: 0.6728
Epoch 3/10
49/49 [=====] - 154s 3s/step - loss: 0.5753 - accuracy: 0.6971 - val_loss: 0.5223 - val_accuracy: 0.7408
Epoch 4/10
49/49 [=====] - 152s 3s/step - loss: 0.5444 - accuracy: 0.7218 - val_loss: 0.5019 - val_accuracy: 0.7564
Epoch 5/10
49/49 [=====] - 153s 3s/step - loss: 0.5217 - accuracy: 0.7390 - val_loss: 0.5017 - val_accuracy: 0.7526
Epoch 6/10
49/49 [=====] - 153s 3s/step - loss: 0.5042 - accuracy: 0.7515 - val_loss: 0.4797 - val_accuracy: 0.7614
Epoch 7/10
49/49 [=====] - 151s 3s/step - loss: 0.4966 - accuracy: 0.7559 - val_loss: 0.4723 - val_accuracy: 0.7702
Epoch 8/10
49/49 [=====] - 152s 3s/step - loss: 0.4901 - accuracy: 0.7611 - val_loss: 0.4692 - val_accuracy: 0.7722
Epoch 9/10
49/49 [=====] - 152s 3s/step - loss: 0.4749 - accuracy: 0.7702 - val_loss: 0.4456 - val_accuracy: 0.7830
Epoch 10/10
49/49 [=====] - 150s 3s/step - loss: 0.4632 - accuracy: 0.7792 - val_loss: 0.4373 - val_accuracy: 0.7924
```

## Bibliografie:

<https://docs.python.org/3/library/tkinter.html>

<https://faroit.com/keras-docs/1.2.0/>

<https://keras.io/>

În realizarea modelului am folosit informații din:

<https://www.youtube.com/watch?v=ENXr1foShrA&t=1s>

<https://colab.research.google.com/>