Animal Species Identification & Emotion Recognition

Table of content

Team Members

Name	*ID*	*Task Sheet*
Nureen Ehab Mahmoud	20221465124	Creating and preprocessing Animals dataset Building the model Presentation
Zainab Mohamed Abdallah	20221310251	Creating and preprocessing Emotions dataset Building the model Report

Project Description

Build one model to apply animal species identification (whether the picture includes cat or dog), and emotions recognition (whether the cat/dog feels happy or sad) on two different datasets that we made by ourselves.

Used tools & algorithms

1. Data Preprocessing

We use *OpenCV* to resize images and normalize their pixel values between 0 and 1.

We use *Keras 'ImageDataGenerator'* for performing image augmentation.

2. Splitting the datasets

We use *train_test_split* to divide the images into train_images and test_images based on specific ratio, and divide the test_images into val_images and test_images again based on the ratio.

3. Building the CNN Model

We create custom data generators to fetch and process data from the specified directoryand sets the number of classes for both animal species and emotion recognition tasks.

We use *Input, Conv2D, MaxPooling2D, Flatten, Dense, Dropout* layers from keras to build CNN neural networks.

We use *Adam optimizer* for model compilation (and set the learning rate to 0.0001)

We defined number of samples and steps per epoch.

We then defined train, val and test generators.

We fit the model using history = model.fit

Then we saved the model.

Then we loaded the model and tested the code.

Expected results

Accuracy of model:

```
0.2528 - emotion_output_loss: 0.1629 - animal_species_output_accuracy: 0.9072 - emotion_output_accuracy: 0.9607Animal batch shape: (32, 224, 224, 3) (32, 2)
0.2513 - emotion_output_loss: 0.1606 - animal_species_output_accuracy: 0.9071 - emotion_output_accuracy: 0.9610Animal batch shape: (32, 224, 224, 3) (32, 2)
0.2483 - emotion_output_loss: 0.1591 - animal_species_output_accuracy: 0.9094 - emotion_output_accuracy: 0.9624Animal batch shape: (32, 224, 224, 3) (32, 2)
0.2469 - emotion_output_loss: 0.1574 - animal_species_output_accuracy: 0.9093 - emotion_output_accuracy: 0.9615 Animal batch shape: (32, 224, 224, 3) (32, 2)
0.2473 - emotion_output_loss: 0.1577 - animal_species_output_accuracy: 0.9081 - emotion_output_accuracy: 0.9618Animal batch shape: (32, 224, 224, 3) (32, 2)
0.2439 - emotion_output_loss: 0.1573 - animal_species_output_accuracy: 0.9091 - emotion_output_accuracy: 0.9620Animal batch shape: (32, 224, 224, 3) (32, 2)
0.2415 - emotion_output_loss: 0.1555 - animal_species_output_accuracy: 0.9120 - emotion_output_accuracy: 0.9622Animal batch shape: (32, 224, 224, 3) (32, 2)
0.2411 - emotion_output_loss: 0.1538 - animal_species_output_accuracy: 0.9148 - emotion_output_accuracy: 0.9625Animal batch shape: (29, 224, 224, 3) (29, 2)
0.2405 - emotion_output_loss: 0.1522 - animal_species_output_accuracy: 0.9155 - emotion_output_accuracy: 0.9636Animal batch shape: (32, 224, 224, 3) (32, 2)
.2418 - emotion_output_loss: 0.1517 - animal_species_output_accuracy: 0.9143 - emotion_output_accuracy: 0.9638 Animal batch shape: (32, 224, 224, 3) (32, 2)
ss: 0.2399 - emotion_output_loss: 0.1518 - animal_species_output_accuracy: 0.9165 - emotion_output_accuracy: 0.9648
```

```
# Print the accuracies
print("Animal Species Accuracy:", history.history['animal_species_output_accuracy'])
print("Emotion Accuracy:", history.history['emotion_output_accuracy'])
```

```
Animal Species Accuracy: [0.5658012628555298, 0.7018800377845764, 0.7719928026199341, 0.8549686670303345, 0.9165170788764954]
Emotion Accuracy: [0.6119133830070496, 0.7599278092384338, 0.8583032488822937, 0.9259927868843079, 0.9648014307022095]
```

Testing the code (2 outputs):

```
1/1 [==============================] - 0s 88ms/step
Predicted Animal Species: Dog
Predicted Emotion: Happy
```

```
1/1 [==============================] - 0s 103ms/step
Predicted Animal Species: Dog
Predicted Emotion: Sad
```