Odd Semester (2023/2024)

BINUS UNIVERSITY INTERNATIONAL

Final Project: Snake Classification

Student Information:		**Surname:**	**Given Name:**	**Student ID Number:**
	1.	Richie	Leonardo	2502005856
	2.	Prasetyo	Jonathan	2501982613
	3.	Faren	Hansel	2501990350

Course Code : COMP6065001 **Course Name :** Artificial Intelligence

Class : L5AC **Lecturer** : Zhandos Yessenbayev, B.Sc., M.Sc., Ph.D.

Type of Assignments: Final Project

Due Date : 17 December 2023

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per the lecturer's instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating, and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity, and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

Declaration of Originality

By signing this assignment, I understand, accept, and consent to BiNus International's terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student:

1. Leonardo Richie
2. Jonathan Prasetyo
3. Hansel Faren

A. Problem description

The project of developing a machine learning snake picture identification involves various hurdles. One of the major issues is assembling a big and diverse library of taxonomically meaningful snake photos. The dataset should emphasize scale patterns because the color of the snakes is problematic for identification. However, gathering such a dataset is time-consuming and necessitates access to a wide range of snake species. Furthermore, the high intra-class variance and low inter-class variation in snake photos make it difficult for machine learning algorithms to effectively identify the images.

The creation of a feature extraction algorithm capable of effectively representing the specific properties of snake photos is a big problem. The approach adopted should be capable of capturing the intricate patterns and textures of snake scales, which can be challenging to depict. Choosing the best machine learning model to categorize the snake images is also a difficulty. The model should be able to handle the distinct aspects of the snake photos as well as the complicated interactions between the components. The chosen model should also be able to handle the imbalanced dataset, as snake photos from some species may be underrepresented. Finally, creating a user-friendly interface for the snake image identifier is a difficulty. The interface should be simple and straightforward, allowing users to rapidly and accurately identify snake species.

B. Solution features, e.g. algorithms used, special technologies used, user interface, etc.

- Algorithms used

- Convolutional Neural Network (CNN): This model is designed based on CNN, a deep learning algorithm commonly used for image classification tasks.
- ImageDataGenerator: This is a utility in TensorFlow for real-time data augmentation during model training. Data augmentation helps to artificially increase the diversity of the training dataset, which can improve model generalization.
- TensorFlow and Keras: TensorFlow is an open source machine learning library, and Keras is a high-level neural network API that runs on top of TensorFlow. In this code, Keras is used to define and train the neural network model.
- CSV Handling with Pandas: The code uses the Pandas library to read and manipulate CSV files (train.csv and test.csv). CSV files are used to map class labels and image file names.

- Data Splitting for Validation: The validation_split parameter in ImageDataGenerator is used to split the dataset into training and validation sets. It is common practice to evaluate model performance on data not seen during training.
- One-Hot Encoding: class_mode='categorical' in flow_from_dataframe indicates that the label is one-hot encoding. This is a common technique in multi-class classification problems.
- Adam Optimizer: The Adam Optimizer is used during model compilation. Adam is an optimization algorithm commonly used to train deep learning models.
- Model Saving: The trained model is saved using the save method in Keras, and the model is saved in HDF5 format with the file name 'snake_classifier_model.h5'.
- Data Augmentation: Data augmentation techniques like rescaling, warping, zooming and horizontal flipping are applied on the training dataset using ImageDataGenerator. This helps improve the model's ability to generalize to different variations of input data.

- User interface

  Our GUI categorizes an uploaded photograph of a snake into one of ten species using a pre-trained deep learning model. The software makes use of the Streamlit library for the user interface and the Keras library for loading the pre-trained model. The software defines a function named "processed_img" that accepts an image path as input, processes the image, and returns the anticipated snake species. The software also defines a "run" function, which generates the user interface and handles image submission and classification.

  The "processed_img" function loads the image using Keras's "load_img" function and resizes it to (224, 224, 3). It then normalizes the pixel values and expands the dimensions to match the pre-trained model's input shape. The pre-trained model is utilized to estimate the class probabilities, and the predicted class is computed using the "argmax" function. Using a dictionary, the anticipated class is then mapped to the correct snake species. Finally, the function outputs the projected snake species.

C. Solution design architecture

1. Data Preparation:

   1.1 Dataset Structure:

The dataset is organized into training and testing sets. Each set is divided into subdirectories, one for each snake class. Image file names and labels are stored in CSV files (train.csv and test.csv).

1.2 Data Augmentation:

Use ImageDataGenerator for data augmentation during training. Augmentation includes rescaling, rotation, width shift, height shift, shear, zoom, and horizontal flip.

1.3 File Path Resolution:

Construct file paths for images using information from CSV files.

2. Model Architecture:

2.1 Convolutional Neural Network (CNN):

Input layer: 224x224x3 image dimensions. Convolutional layers with ReLU activation and max-pooling. Flatten layer to transition from convolutional to dense layers. Dense layers with ReLU activation for feature extraction. Output layer with softmax activation for multi-class classification.

2.2 Layers Details:

Conv1: 64 filters, 3x3 kernel, ReLU activation. MaxPool1: 2x2 max-pooling. Conv2: 128 filters, 3x3 kernel, ReLU activation. MaxPool2: 2x2 max-pooling. Conv3: 256 filters, 3x3 kernel, ReLU activation. MaxPool3: 2x2 max-pooling. Flatten: Flatten the output for dense layers. Dense1: 512 neurons, ReLU activation. Dense2: Output layer with neurons equal to the number of classes, softmax activation.

2.3 Additional Components:

Batch Normalization after convolutional layers. Dropout layer before the final dense layer to reduce overfitting.

3. Training:

3.1 Compile the Model:

Use the Adam optimizer. Categorical cross entropy loss for multi-class classification. Metrics include accuracy.

3.2 Data Generator:

Use data generators for efficient memory usage. Training and validation generators created using ImageDataGenerator and flow_from_dataframe.

3.3 Callbacks:

Learning rate scheduler to adjust the learning rate during training. Early stopping to prevent overfitting.

3.4 Hyperparameters:

Number of epochs: 20 Batch size: 64

3.5 Class Weighting:

Optionally use class weighting to handle class imbalance.

4. Evaluation:

4.1 Model Evaluation:

After training, evaluate the model on a test set using the evaluate method. Print test accuracy and other relevant metrics.

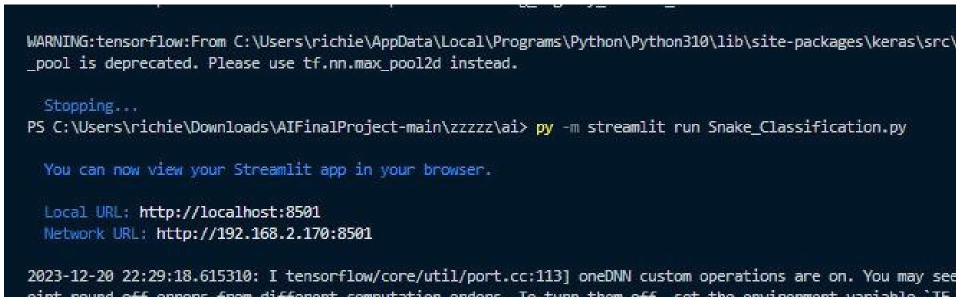
D. Experiments or tests that you have done.

There might be lots of animals that we can make as our project but we need to choose one animal. That conclusion was down to choosing snakes as our dataset and identifying the snake images for our project. We have experiments using some models to train them with the snake dataset but it comes down to using vgg16 or CNN. There are some trial and error where the training process is too long or being cut down when running on kaggle's notebook because it will stop automatically if there is no activity for 12 hours.

After the process of training the model is done, we then save the trained model and try to apply it on the GUI code so the user can use the application. There's not that going on in this part because we need to use the trained model and apply it to the application. Some obstacles that we face is when sizing the images because you don't know what size of the images that the user might upload to the application.

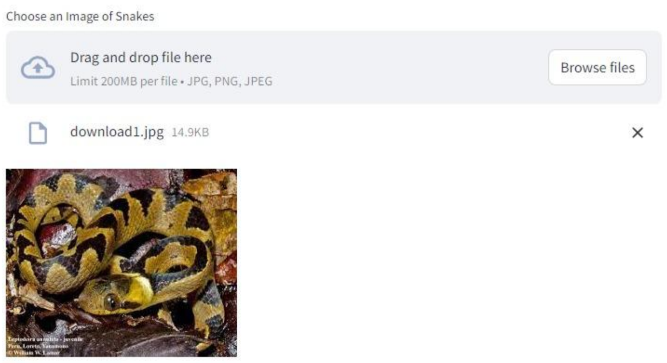
E. Program manual (with screenshots)

a. Download files from this link: https://github.com/LeonardoRichie/AIFinalProject/tree/main/ai
b. Run the python file first.
c. Open terminal and write this command.

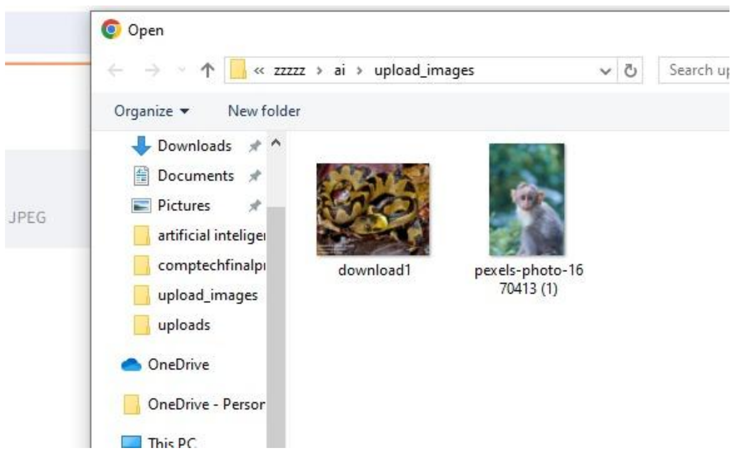


```
WARNING:tensorflow:From C:\Users\richie\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\src\
_pool is deprecated. Please use tf.nn.max_pool2d instead.

  Stopping...
PS C:\Users\richie\Downloads\AIFinalProject-main\zzzzz\ai> py -m streamlit run Snake_Classification.py

  You can now view your Streamlit app in your browser.

  Local URL: http://localhost:8501
  Network URL: http://192.168.2.170:8501

2023-12-20 22:29:18.615310: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see
```

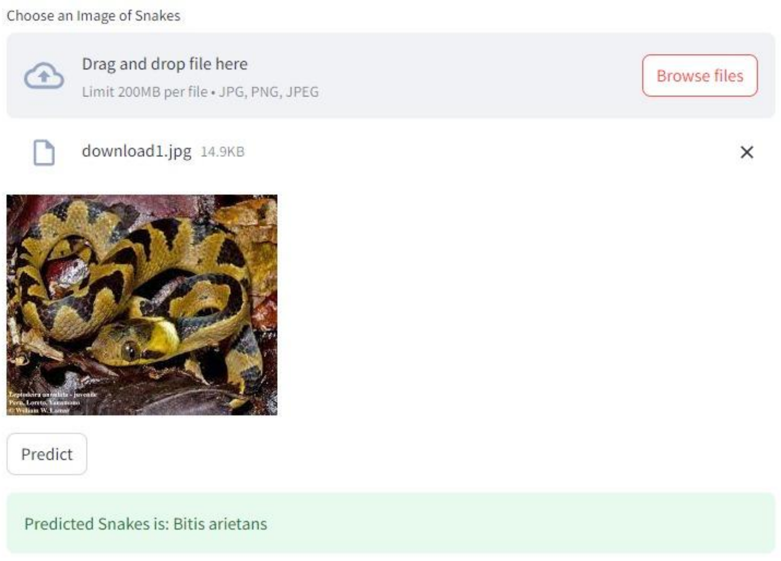
d. It will show this UI after running it.


Choose an Image of Snakes
Drag and drop file here
Limit 200MB per file • JPG, PNG, JPEG
Browse files
download1.jpg 14.9KB

e. Make a folder called “upload_images” and put the picture you want to test.



f. Then click on the predict button and it will show results.


Choose an Image of Snakes

Drag and drop file here

Limit 200MB per file • JPG, PNG, JPEG

download1.jpg 14.9KB

Predicted Snakes is: Bitis arietans

F. Link to video of the application demo (with max. length of 2 minutes)

Here is to our video demo:
https://drive.google.com/drive/folders/1dMTqllHhp2JmkKBdA6211Y4JQRfTsRrd?usp=drive_link

G. Link to your GIT repository

This is the link to our project github repository:
https://github.com/LeonardoRichie/AIFinalProject