Deep Learning – Case Study

Title: Snake Breed Prediction

Name: Deep Modi

Enrollment Number: 18012011016

Batch: DL1

Guided by: Prof Ketan J. Sarvakar

Introduction

In this project, we will build and train convolutional neural network for classifying images of Snake. The input image will be analysed and then the output is predicted. The model that is implemented can be extended to a website or any mobile device as per the need. The Snake dataset used in this project is available on Kaggle.

Notebook Used

  Google Colab - Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

Tools and Libraries

Libraries used are numpy, pandas, keras, sklearn, maplotlib

a. Numpy: NumPy offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more.

b. Pandas: panda is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

c. Keras: Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load.

d. Sklearn: Sklearn has Simple and efficient tools for predictive data analysis, accessible to everybody, and reusable in various contexts built on NumPy, SciPy, and matplotlib.

e. Maplotlib: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.

Model Explanation:-

CNN
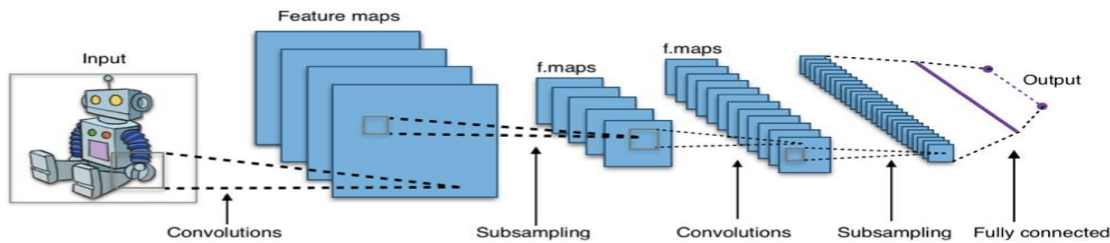
Convolution Neural Network

Source: Wikipedia

Model Architecture:-

```
Model: "sequential"
_____
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 89, 89, 16)        208

conv2d_1 (Conv2D)            (None, 44, 44, 32)        2080

conv2d_2 (Conv2D)            (None, 43, 43, 32)        4128

dropout (Dropout)            (None, 43, 43, 32)        0

conv2d_3 (Conv2D)            (None, 21, 21, 32)        4128

dropout_1 (Dropout)          (None, 21, 21, 32)        0

conv2d_4 (Conv2D)            (None, 10, 10, 32)        4128

flatten (Flatten)            (None, 3200)              0

dense (Dense)                (None, 128)               409728

dropout_2 (Dropout)          (None, 128)               0

dense_1 (Dense)              (None, 64)                8256

dropout_3 (Dropout)          (None, 64)                0

dense_2 (Dense)              (None, 35)                2240

=================================================================
Total params: 434,896
Trainable params: 434,896
Non-trainable params: 0
_____
```

Deep Learning – Case Study

Workflow:-

Step 1: Import Requires Libraries

Step 2: Get the data

Step 3: Split the data into train and test

Step 4: Split features from labels Train and test

Step 5: Normalize the data

Step 6: Build the model

Step 7: Inspect the model

Step 8: Train the model for 10 epochs

Step 9: Result Prediction

Output:-

```
[20] cnn_model.fit_generator(  train_data_with_label, steps_per_epoch=steps_per_epoch,epochs = 10, shuffle=  True )

    /usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated a
      """Entry point for launching an IPython kernel.
    Epoch 1/10
    6/6 [==============================] - 72s 12s/step - loss: 3.5487 - categorical_accuracy: 0.0349
    Epoch 2/10
    6/6 [==============================] - 73s 12s/step - loss: 3.4807 - categorical_accuracy: 0.0603
    Epoch 3/10
    6/6 [==============================] - 72s 11s/step - loss: 3.4327 - categorical_accuracy: 0.0890
    Epoch 4/10
    6/6 [==============================] - 73s 11s/step - loss: 3.4045 - categorical_accuracy: 0.0822
    Epoch 5/10
    6/6 [==============================] - 72s 11s/step - loss: 3.3933 - categorical_accuracy: 0.0864
    Epoch 6/10
    6/6 [==============================] - 86s 14s/step - loss: 3.3895 - categorical_accuracy: 0.0777
    Epoch 7/10
    6/6 [==============================] - 73s 11s/step - loss: 3.3815 - categorical_accuracy: 0.0871
    Epoch 8/10
    6/6 [==============================] - 72s 11s/step - loss: 3.3785 - categorical_accuracy: 0.0868
    Epoch 9/10
    6/6 [==============================] - 76s 12s/step - loss: 3.3756 - categorical_accuracy: 0.0877
    Epoch 10/10
    6/6 [==============================] - 84s 13s/step - loss: 3.3737 - categorical_accuracy: 0.0911
    <keras.callbacks.History at 0x7f1937963a90>
```

Conclusion:-

Model is 91% Accurate.

Deep Learning – Case Study

Dataset :-

`https://www.kaggle.com/manjunathns/identifying-snake-bread-ann`

Project Link :-

https://github.com/Deepmodi1301/DL_Project/blob/main/identifying-snake-bread-ann%20(2).ipynb