Image Classification

Business Objective

Image classification helps to classify a given set of images as their respective category classes. There are many applications of image classification today, one of them being self-driving cars. An image classification model can be built that recognizes various objects, such as vehicles, people, moving objects, etc., on the road to enable autonomous driving.

There are three main classes of input images in this project, and we need to build a model that can correctly identify a given image. To achieve this, we will be using one of the famous machine learning algorithms used for image classification, i.e., Convolutional Neural Network (or CNN).

Data Description

We will be using a dataset of images categorized into three types, namely the driving license, social security, and others.
The training and testing data each contains these three subfolders. There are around 50 images in each subfolder of testing data, while approximately 200 images in each subfolder of training data.

Aim

To build a sequential model that can perform multiclass classification on a given set of data images.

Tech stack

- Language - Python
- Libraries - numpy, matplotlib, tensorflow, cv2

Approach

1. Importing the required libraries.
2. Load and read the data images
3. Data Visualization
    - Count plot
4. Data pre-processing
    - Create train and validation data
    - Normalize the data
    - Reshape the data images
5. Data augmentation
    - Using ImageDataGenerator
6. Model Training
    - Create a sequential model
    - Add convolution, maxpool,dropout layers
    - Add the softmax activation function (As this is a multiclass classification problem)
    - Pass the optimizer parameter
    - Compile the model
    - Fit and train the model
    - Check for the predictions
    - Save the model in h5 format.
7. Inferencing the model
    - Prediction on the test data

Modular code overview

```
input
  |_Data
     |_Training_data
     |_Testing_data


src
  |_Engine.py
  |_ML_Pipeline
            |_admin.py
            |_inference.py
            |_training.py


lib
  |_training_classification.ipynb
  |_Inference.ipynb


output
  |_model.h5
```

Once you unzip the modular_code.zip file, you can find the following folders within it.

1. Input
2. Source Folder
3. Output
4. Lib

1. Input folder **-** It contains all the data that we have for analysis. We have two folders. Training_data and testing_data. The following three subfolders are present
    - driving_license
    - social_security
    - others

2. Source folder - This is the most important folder of the project. This folder contains all the modularized code for all the above steps in a modularized manner. This folder consists of:
    - Engine.py
    - ML_Pipeline
      The ML_Pipeline is a folder that contains all the functions put into different python files, which are appropriately named. These python functions are then called inside the engine.py file.

3. Output folder **-** The output folder contains the fitted model that we trained for this data. This model can be easily loaded and used for future use, and the user need not have to train all the models from the beginning.

4. Lib - This folder contains two notebooks.
    - training_classification.ipynb
    - Inference.ipynb

    There is a ppt present in the reference subfolder. This is the one used during the presentation.

Project Takeaways

1. How the human brain recognizes objects
2. Basics of convolutional neural network
3. Input images to CNN
4. Architecture of CNN
5. Convolution in CNN
6. Activation functions - Step, Sigmoid, ReLU, leaky ReLU
7. Pooling layers - max and average pooling
8. Flattening in CNN
9. Multiclass and multilabel classification
10. Sigmoid and softmax functions
11. Loss function - binary cross-entropy
12. Optimizers - gradient descent, stochastic gradient descent
13. Data augmentation
14. Model building and training
15. Save the model
16. Predictions on the test data