Build a CNN Model with PyTorch for Image Classification

Overview

Convolutional Neural Network(CNN) is a deep learning algorithm that learns directly from data, eliminating the need for manual feature extraction. CNNs are particularly useful for the image data which helps in finding patterns in images to recognize objects.
In this project, we will build a CNN model for image classification where images will be classified into classes of social security cards, driving licenses, and others. We have used PyTorch for building the model which is different than other deep learning frameworks as it uses dynamic computational graphs. Also, PyTorch is pythonic in nature and it is built over NumPy which makes it easy to use.
If you haven't visited already, here is the previous project of the series Learn How to Build a Logistic Regression Model in PyTorch.

Aim

- To understand the working of CNN
- To build a Convolutional Neural Network model to classify images into different classes in PyTorch

Tech Stack

- ➔ Language: Python
- ➔ Libraries: pytorch, pandas, matplotlib, numpy, opnecv_python_headless, torchvision

Data Description

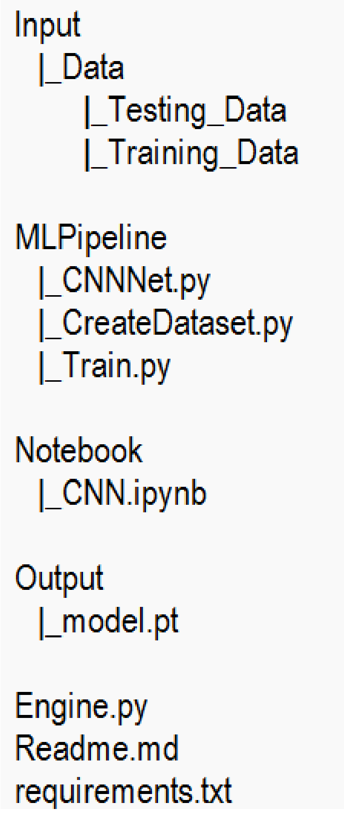
Dataset used in this project are images of driving license, social security, and others categorized into respective categories. The images are of different shapes and sizes which are preprocessed before modeling.

Approach

- Data Loading
- Data Preprocessing
  - ○ Resizing and scaling of the images
  - ○ Encoding of the class labels

- Model Building and Training
  - CNN model building in PyTorch

Modular Code Overview



```
Input
  |_Data
      |_Testing_Data
      |_Training_Data

MLPipeline
  |_CNNNet.py
  |_CreateDataset.py
  |_Train.py

Notebook
  |_CNN.ipynb

Output
  |_model.pt

Engine.py
Readme.md
requirements.txt
```

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

1. Input
2. ML_Pipeline
3. Notebook
4. Output
5. Engine.py
6. Readme.md
7. requirements.txt

1. The Input folder contains the data that we have for analysis. In our case, it contains training data and testing of images to be classified

2. The Notebook folder contains the jupyter notebook file of the project

3. 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

4. The Output folder contains the saved CNN model

5. The requirements.txt file has all the required libraries with respective versions. Kindly install the file by using the command **pip install -r requirements.txt**

6. **All the instructions for running the code are present in Readme.md file**

Takeaways

1. What is PyTorch?
2. PyTorch vs Tensorflow
3. What is computer vision?
4. What is Convolutional Neural Network(CNN)?
5. What is the use of CNN?
6. The architecture of CNN
7. How to extract features from images
8. What are kernels in CNN?
9. What are padding and poolings?
10. What is the use of Pooling and Padding?
11. Types of pooling
12. Mathematic of Convolutional Neural Network
13. Data loading in PyTorch
14. Data Preprocessing in PyTorch
15. CNN implementation in PyTorch