Visual Analytics Portfolio

Assignment 4: Self-assigned project - classifying fruits with VGG16

Cultural Data Science, 2023

Author: Aleksander Moeslund Wael
Student no. 202005192

About the project

This repo contains code for conducting image classification on a dataset of fruit images. Two models are fit to the data, a simple sequential model and a large retrained CNN model (VGG16). The project demonstrates the

Data

The dataset used for the project is the Fruit Classification dataset from Kaggle. This dataset consists of 22495 images of fruits across 33 classes (fruit types). Images are 100x100 resolution and masked to isolate the fruits. The samples in each class appear to be quite heterogenous in this dataset, so a high accuracy score is expected when classifying the images.

Model

Image classification was performed using two approaches:

1. Initiating, training and predicting with a simple sequential model.
2. Finetuning and predicting with the `VGG16` model.

The `VGG16` model is 16 layers deep, and has approx 138 million parameters and is trained on the ImageNet dataset. (source).

Both models are handled in the TensorFlow framework.

Pipeline

There are two Python scripts in the `src` folder, `simple_fruit_classifier.py` and `fruit_classifier.py`, which contain code pipelines for performing image classification using the two models. Each script follows these steps:

1. Import dependencies
2. Load and preprocess data
3. Setup data generators
4. Setup model
5. Fit model to data
6. Plot and save learning curves
7. Print and save classification report

The `fruit_classifier.py` script uses the `VGG16` model.

Requirements

The code is tested on Python 3.11.2. Futhermore, if your OS is not UNIX-based, a bash-compatible terminal is required for running shell scripts (such as Git for Windows).

Usage

The repo was setup to work with Windows (the WIN_ files), MacOS and Linux (the MACL_ files).

1. Clone repository to desired directory

```
git clone https://github.com/alekswael/assignment4-self-assigned-project
cd assignment4-self-assigned-project
```

2. Run setup script

NOTE: Depending on your OS, run either `WIN_setup.sh` or `MACL_setup.sh`.

The setup script does the following:

1. Creates a virtual environment for the project
2. Activates the virtual environment
3. Installs the correct versions of the packages required
4. Deactivates the virtual environment

```
bash WIN_setup.sh
```

3. Run pipeline

NOTE: Depending on your OS, run either `WIN_run.sh` or `MACL_run.sh`.

Run the script in a bash terminal.

The script does the following:

1. Activates the virtual environment
2. Runs `cnn_fashion.py` located in the `src` folder
3. Deactivates the virtual environment

```
bash WIN_run.sh
bash WIN_run_simple.sh
```

Note on model tweaks

Some model parameters can be set through the `argparse` module. However, this requires running the Python script seperately OR altering the `run*.sh` file to include the arguments. The Python script is located in the `src`

folder. Make sure to activate the environment before running the Python script.

```
simple_fruit_classifier.py [-h] [-bs BATCH_SIZE] [-e EPOCHS]

options:
  -h, --help            show this help message and exit
  -bs BATCH_SIZE, --batch_size BATCH_SIZE
                        Batch size for training. (default: 32)
  -e EPOCHS, --epochs EPOCHS
                        Number of epochs to train for. (default: 100)
```

```
fruit_classifier.py [-h] [-bs BATCH_SIZE] [-e EPOCHS]

options:
  -h, --help            show this help message and exit
  -bs BATCH_SIZE, --batch_size BATCH_SIZE
                        Batch size for training. (default: 32)
  -e EPOCHS, --epochs EPOCHS
                        Number of epochs to train for. (default: 100)
```

Repository structure

This repository has the following structure:

```
│   MACL_run.sh
│   MACL_run_simple.sh
│   MACL_setup.sh
│   README.md
│   requirements.txt
│   WIN_run.sh
│   WIN_run_simple.sh
│   WIN_setup.sh
│
├───fruits_v2
│   ├───test
│   │   ├───Apple Braeburn
│   │   │       Apple Braeburn.jpg
│   │   │       ...
│   │   ├───Apple Granny Smith
│   │   │       Apple Granny Smith.jpg
│   │   │       ...
│   │   ...
│   ├───train
│   │   ├───Apple Braeburn
│   │   │       Apple Braeburn.jpg
│   │   │       ...
│   │   ├───Apple Granny Smith
│   │   │       Apple Granny Smith.jpg
```

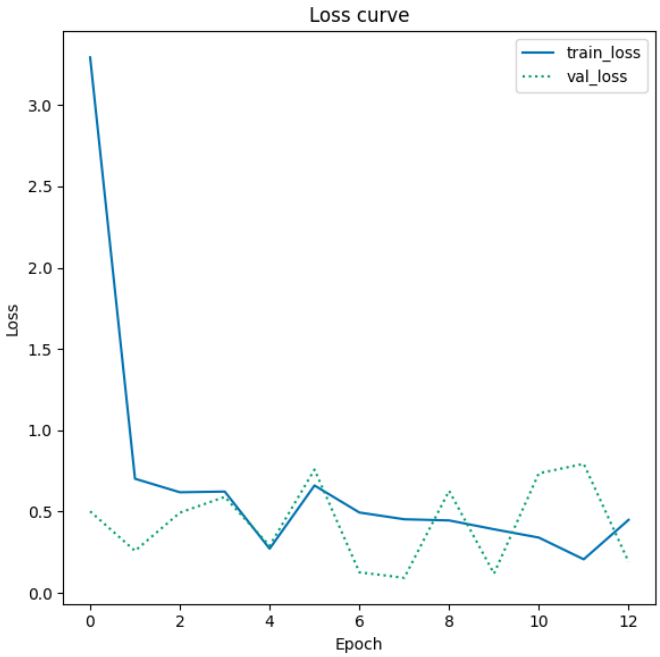
```
|   |   |       ...
|   |   ...
|   ├──val
|   |   ├──Apple Braeburn
|   |   |       Apple Braeburn.jpg
|   |   |       ...
|   |   ├──Apple Granny Smith
|   |   |       Apple Granny Smith.jpg
|   |   |       ...
|   |   ...
|
├──out
|       classification.png
|       classification_report.txt
|       simple_classification.png
|       simple_classification_report.txt
|
└──src
        fruit_classifier.py
        simple_fruit_classifier.py
```

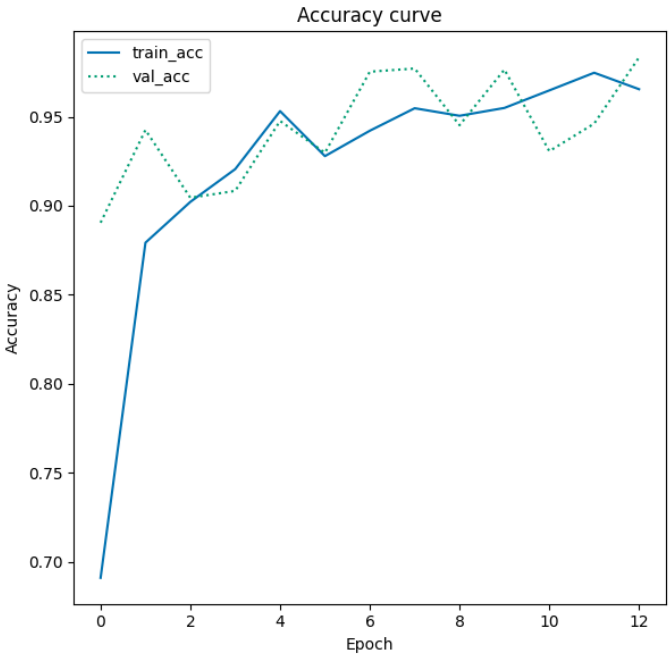
Remarks on findings

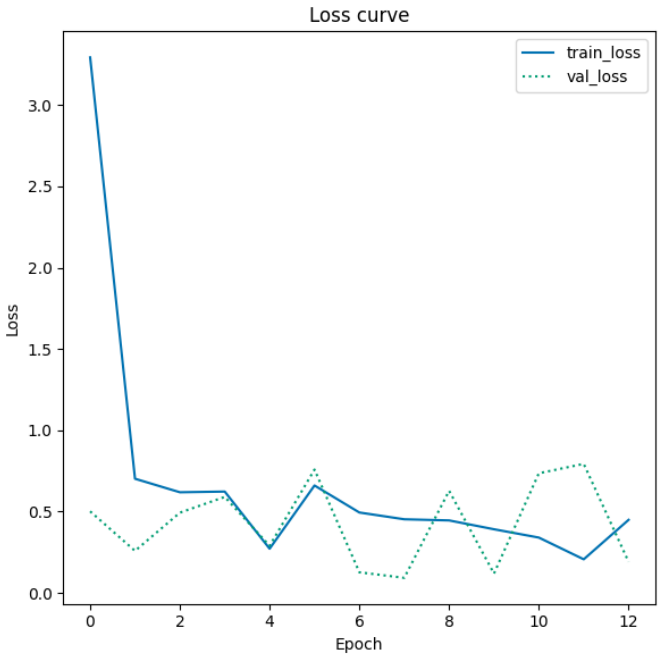
	precision	recall	f1-score	support
Apple Braeburn	1.00	0.80	0.89	50
Apple Granny Smith	1.00	0.98	0.99	50
Apricot	1.00	1.00	1.00	50
Avocado	1.00	1.00	1.00	44
Banana	1.00	1.00	1.00	49
Blueberry	1.00	1.00	1.00	47
Cactus fruit	0.94	1.00	0.97	49
Cantaloupe	1.00	1.00	1.00	50
Cherry	1.00	1.00	1.00	50
Clementine	1.00	1.00	1.00	49
Corn	1.00	1.00	1.00	45
Cucumber Ripe	0.91	1.00	0.95	40
Grape Blue	1.00	1.00	1.00	100
Kiwi	1.00	0.91	0.96	47
Lemon	1.00	1.00	1.00	50
Limes	0.98	1.00	0.99	49
Mango	1.00	1.00	1.00	49
Onion White	1.00	1.00	1.00	45
Orange	1.00	1.00	1.00	49
Papaya	1.00	1.00	1.00	50
Passion Fruit	1.00	1.00	1.00	49
Peach	0.81	1.00	0.89	50
Pear	0.99	0.99	0.99	70
Pepper Green	1.00	1.00	1.00	46
Pepper Red	0.99	1.00	0.99	67
Pineapple	1.00	1.00	1.00	49

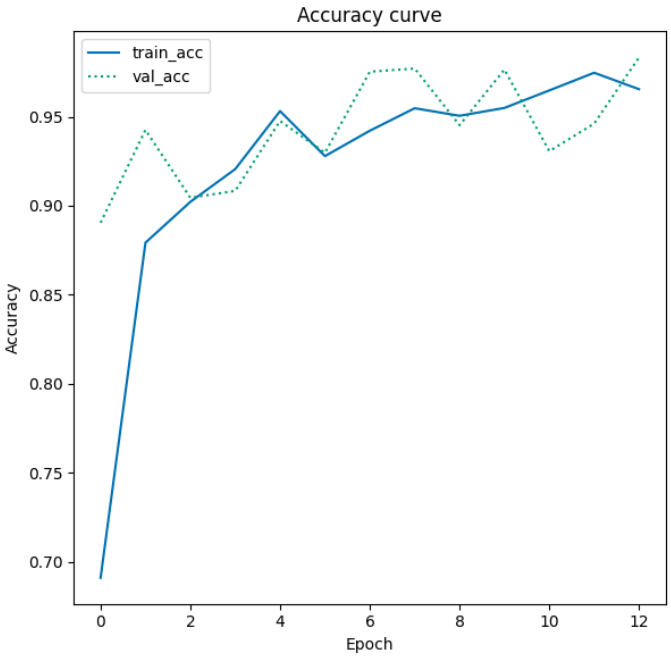
```
                Plum       1.00      1.00      1.00        46
          Pomegranate       1.00      0.94      0.97        50
          Potato Red        1.00      0.91      0.95        45
           Raspberry        1.00      1.00      1.00        49
          Strawberry        1.00      1.00      1.00        50
              Tomato        0.99      1.00      0.99        75
          Watermelon        1.00      1.00      1.00        48

            accuracy                            0.99      1706
           macro avg        0.99      0.99      0.99      1706
        weighted avg        0.99      0.99      0.99      1706
```






Loss curve
train_loss
val_loss
Loss
Epoch
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
2
4
6
8
10
12


Accuracy curve
train_acc
val_acc
Accuracy
Epoch
0.70
0.75
0.80
0.85
0.90
0.95
0
2
4
6
8
10
12