CSC1028 AutoML Test Plan

Mark Jardine

40291581

mjardine03@qub.ac.uk

Overview of Project Specification

AutoML (formalising and automating the stages in training a Machine Learning algorithm)*

Developing machine learning models to solve problems is often described as more of an art than a science. This is another way of saying that it is very hard to guess how well a given technique will work on a problem. AI developers therefore spend a lot of their time searching over different machine learning architectures and hyper parameters trying to find ones that work well for their problem. This can lead to messy code and create a feeling like you are building on sand, as soon as you decide that one part of the model is working well changing another part may invalidate that choice. This has led to the field of AutoML which is a process of automatically searching for good AI architecture and hyper parameters. This approach is a good way to approach AI problems where instead of writing a single algorithm that you change you develop a python script that can generate different AI experiments with different architectures and parameters. Using this approach you can automatically run lots of different experiments and automatically search for the best approach. As you work on the project you are adding different alternative architectures to your python script so that you can automatically search over more approaches. In this way you never take a step back, every choice you add is at worst making your system try unhelpful approaches but the optimal design can be easily produced from your configuration.

The goal of this project is to take one or more of the AI algorithms described as part of the FastAI course (the leading course on Deep learning which includes a number of state of the art models for solving common machine learning problems). For each problem express the choices available in how the algorithm is implemented using a csv format (like a row in excel).

Make the resulting training algorithms capable of running on google colab and as a docker gpu container so that the machine learning algorithms can be easily trained on a cloud server.

Multiple students can work on this project each focused on a different AI problem being configured e.g. Image recognition, time series prediction, object detection etc.

Requirements

- Python 3 (I used 3.8.6)
- Pip (I used 21.0.1)

Setup

- Download the “venv_for_user_download” folder from here.
- Run the setup script to create a local python virtual environment and install dependencies etc.

Test Case 1

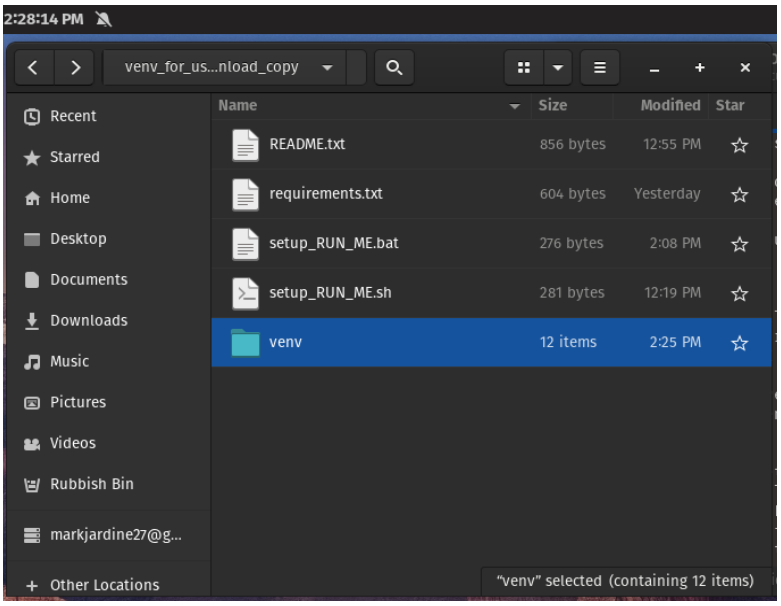
Running the setup script (shell script)

User action

User runs ***setup_RUN_ME.sh***

Expected output

Venv folder is created with all dependencies present and files moved to correct locations.



Test Case 2

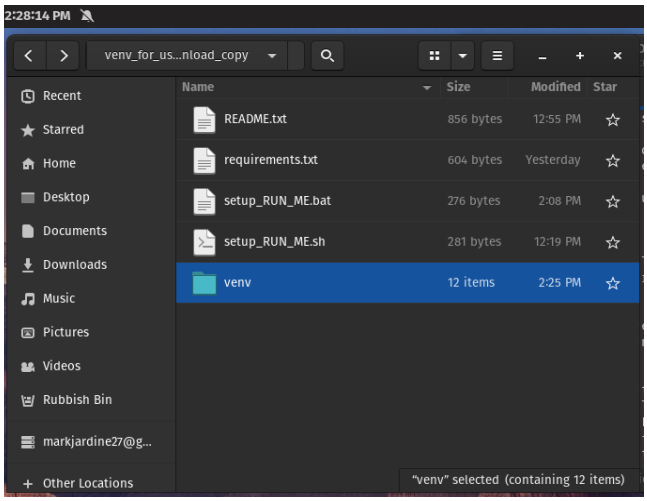
Running the setup script (batch)

User action

User runs ***setup_RUN_ME.bat***

Expected output

Venv folder is created with all dependencies present and files moved to correct locations.



Test Case 3

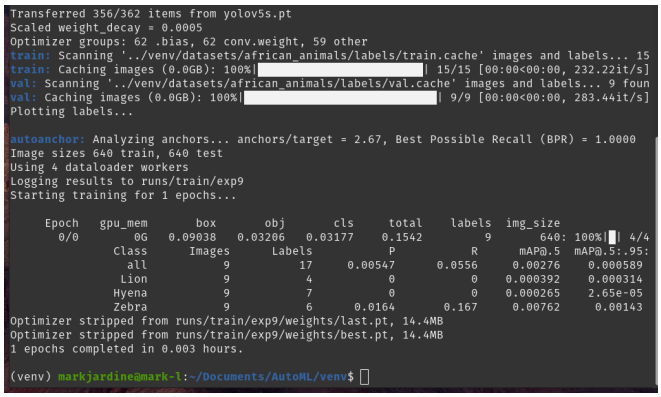
Running the train script (shell script)

User action

User runs ***train.sh***

Expected output

Train.py runs on script-specified parameters.



Test Case 4

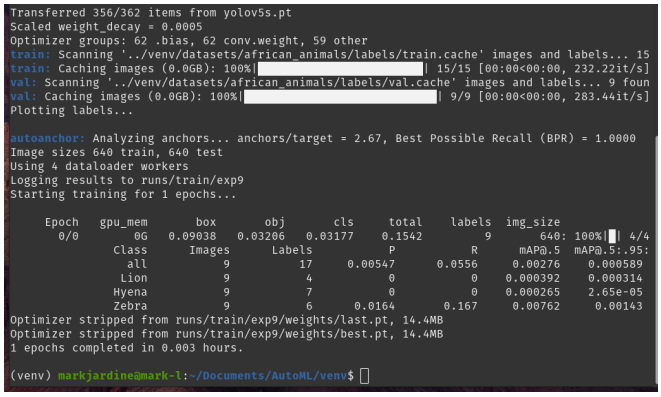
Running the train script (batch)

User action

User runs ***train.bat***

Expected output

Train.py runs on script-specified parameters.



Test Case 5

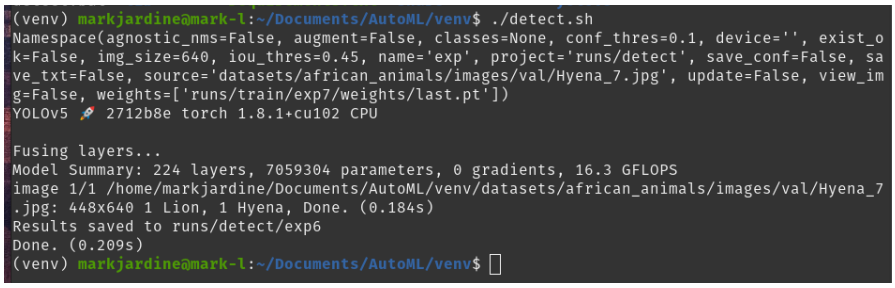
Running the detect script (shell script)

User action

User runs ***detect.sh***

Expected output

Detect.py runs on script-specified parameters.



Test Case 6

Running the detect script (batch)

User action

User runs ***detect.bat***

Expected output

Detect.py runs on script-specified parameters.