

DOCUMENTATION OF A SIMPLE AI PROJECT

By: Kotryna Venskūnaitė

Professor: Vytautas Barzdaitis

This documentation aims to explain a simple AI project I did for VDU university.

SUMMARY OF THE PROJECT

The purpose of this project is to distinguish between two fictional characters named Denji and Aki (from "Chainsawman"). The AI takes two folders filled with pictures with each of their faces from the manga and after training itself can give a percentage of how much any given picture looks more like one than the other.



Denji

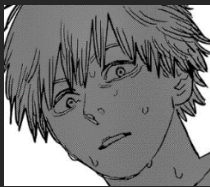


Aki

EXAMPLES AND IN DEPTH EXPLANATION

In “load_AI.py” the program takes a png file from the destination pointed at in line 10, then, after processing the file, gives an estimate percentage on how much an image resembles one of the characters. If an image scores 85% - 100% then it thinks it's Denji, if it gives an estimate of around 20% - 35% off from the 50% mark, then it considers the image to be Aki, then 20% - 0% is given to images that can not be identified to either of them.

EXAMPLES WITH A MODEL OF 250 EPOCHS:



0.01% - Unrelated
99.99% - Denji.



0.00% - Unrelated
100.00% - Denji.



57.17% - Unrelated
42.83% - Denji.



98.26% - Unrelated
1.74% - Denji.

It is important to note that “Aki” produces very poor results compared to “Denji”. This is because of many reasons like:

- The models score is biased towards “Denji” (his score is in one of the extreme sides of the spectrum)
- The data pool is quite low, both consisting of less than 70 images each. The standard is usually a couple hundred/thousand images.
- The model puts an importance on black to white color ratio that makes identifying “Denji” way easier, thus the high disparity of percentages between images of “Aki”. This can also be applied to unrelated images too as pictures with a lot of white are seen as Denji and pictures with a high Black ratio on the upper half are seen as mixed (Aki)

SHOWCASE OF FAULTY RECOGNITION AND UNRELATED IMAGES



61.85% - Unrelated
38.15% - Denji.



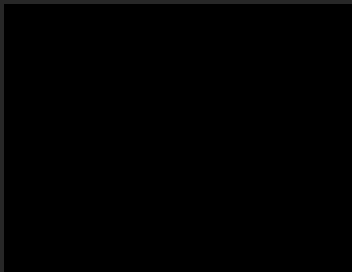
0.22% - Unrelated
99.78% - Denji.



1.65% - Unrelated
98.35% - Denji.



64.86% - Unrelated
35.14% - Denji.



99.98% - Unrelated
0.02% - Denji.



0.00% - Unrelated
100.00% - Denji.



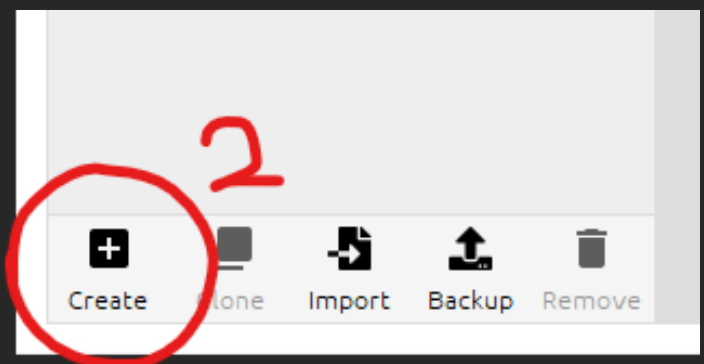
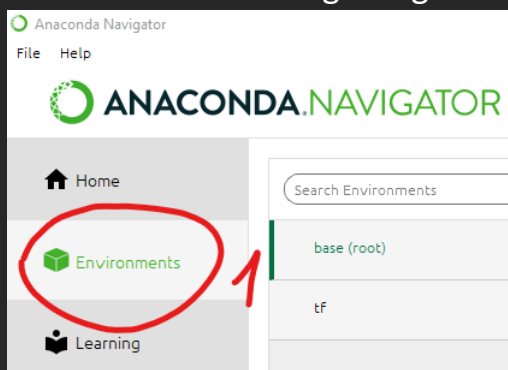
16.37% - Unrelated
83.63% - Denji.



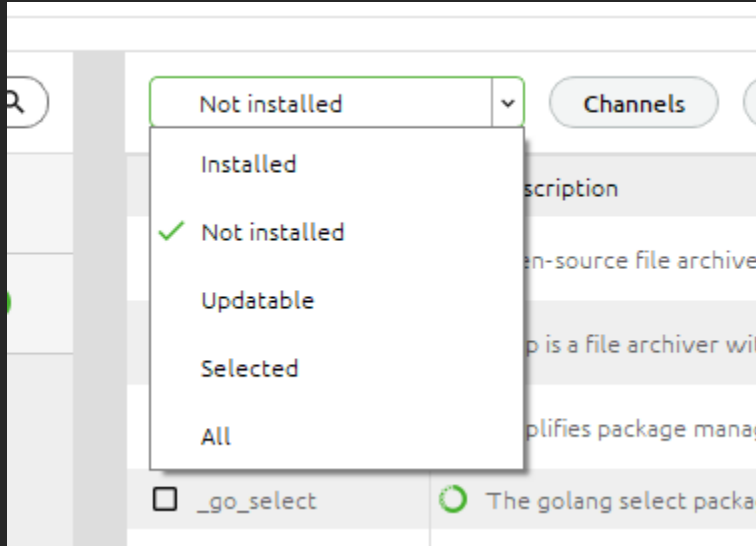
66.31% - Unrelated
33.69% - Denji.

STEP BY STEP INSTALLATION

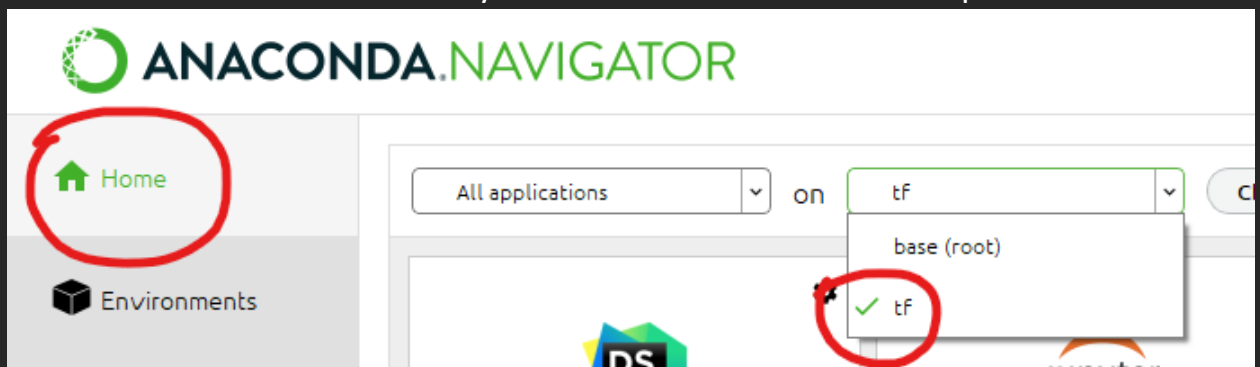
1. Download “Python”, “Anaconda” and “Visual Studio Code” latest versions.
2. In Anaconda navigator go to “Environments” and click “Create”.



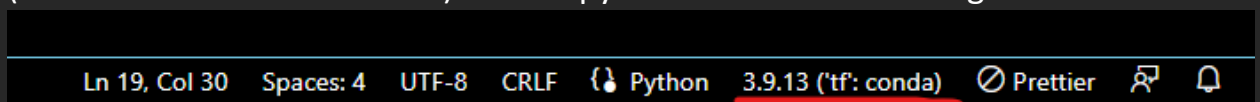
3. Name your new environment and make sure “Python” is selected in the “Packages” section with the latest version selected.
4. In the upper left corner in the drop down menu select “Not installed”



5. Search and install these packages: “Tensorflow”, “Keras”, “Pydot” and “Pillow”.
6. Go to back to home and select your new environment in the top left corner.



7. Launch VS Code. When launched, it should specify that it is running in (“environment name”:conda) next to python version in lower right corner.



8. (Optional) if you want to change your soon to be created model path you can do so in “Create_AI_model.py” line 110 and “Load_AI.py” line 5. Otherwise it will create new folders in the directory the code is running from.
9. First run “Create_AI_model.py” to create a model. You can adjust the “epoch” number to a lower one (like 25) so that it would take less time (but worse results). 1 epoch takes 10s – 15s (depending on your pc).
10. In “Load_AI.py” line 10 change the string to your desired image name and run the program. In the terminal you will get your results.

EXPLANATION OF THE TOOLS AND LIBRARIES USED

TensorFlow – a Python-friendly open source library for numerical computation. It eases the process of acquiring data, training models, serving predictions, and refining future results. It was created by Google Brain.

Keras - an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable error messages. It is built on top of TensorFlow 2.



Anaconda Navigator - a desktop graphical user interface (GUI) included in Anaconda® Distribution that allows you to launch applications and manage conda packages, environments, and channels without using command line interface (CLI) commands.

Epoch - An epoch is when all the training data is used at once and is defined as the total number of iterations of all the training data in one cycle for training the machine learning model.

Another way to define an epoch is the number of passes a training dataset takes around an algorithm. One pass is counted when the data set has done both forward and backward passes.

The number of epochs is considered a hyperparameter. It defines the number of times the entire data set has to be worked through the learning algorithm.

SHOWCASE OF DIFFERENT NUMBER EPOCHS IM THE MODEL

	250	150	50	25
	0.01% - Unrelated 99.99% - Denji.	30.43% - Unrelated 69.57% - Denji.	32.24% - Unrelated 67.76% - Denji.	43.63% - Unrelated 56.37% - Denji.
	40.61% - Unrelated 59.39% - Denji.	78.28% - Unrelated 21.72% - Denji.	41.12% - Unrelated 58.88% - Denji.	49.09% - Unrelated 50.91% - Denji.

This is my first ever AI project and I got to learn a lot of new concepts and skills that I hope to refine and improve in the future.