**LAB REPORT**

Gemini

TEAM MEMBERS

Ray Espinoza

Ricky Ao

Tran Huynh

Contents

[PURPOSES AND GOALS 3](#_heading=h.gjdgxs)

[HOW TO INSTALL THE PROGRAMS](#_heading=h.30j0zll) 3

[DEPENDENCIES](#_heading=h.1fob9te) 3

[HOW TO RUN THE PROGRAMS](#_heading=h.3znysh7) 3

[DESIGN ARCHITECTURE](#_heading=h.2et92p0) 3

[PROCESS & WORKFLOW](#_heading=h.tyjcwt) 4

[TEST DATA](#_heading=h.3dy6vkm) 5

[VIDEO RECORDINGS](#_heading=h.1t3h5sf) 11

[CONCLUSIONS](#_heading=h.4d34og8) 11

[REFERENCES](#_heading=h.2s8eyo1) 12

# PURPOSES AND GOALS

The purpose of this lab is to use Convolutional Neural Network (CNN) to recognize color images and use different methods to improve the accuracy of the code. The first part of the lab is to use the CNN baseline code to improve the accuracy of 70% to 90% and recognize successfully 90% of the given test images. We will use three methods: increasing dropout, data augmentation, and batch normalization to improve the accuracy of the code.

The next part of the lab is to add three unrecognizable images to the test images code and successfully recognize them.

The third part of the lab will consist of the game development and adding hacks and tweaks to the game.

After adding hacks and tweaks to the game, we will use Python to perform a hello world with openAI. The goal is to change the picture of the website from a dog to something else, change the title from “Name my pet“ to something more applicable, and for the last part we will change the code to generate four names instead of 3 names.

The last part of the lab will consist of running the given ChatGPT code and making changes to the model\_engine, prompt, max\_tokens, number of n, stop time, and temperature. Then we will document any output difference after each change.

# HOW TO INSTALL THE PROGRAMS

## DEPENDENCIES

In order for the Python code to work we will need to pip install several dependencies. You will need to pip install the following: pip install TensorFlow, pip install Keras, pip install h5py, pip install Matplotlib, and pip install numpy.

# HOW TO RUN THE PROGRAMS

To execute the CNN baseline and test\_image program successfully, you must begin by importing the essential libraries and datasets. The Cifar dataset comprises 60,000 32x32 color images distributed across 10 distinct classes. This dataset is divided into various training batches and a singular test batch, each containing a specific number of images. In the context of this program, we're working with approximately 1563 images, with the test batch being processed roughly 30 times. Upon running the CNN baseline code, the program will conclude by saving the model to a .h file. This .h file is subsequently loaded into the test\_image code, facilitating the validation of images and the prediction of their respective labels.

In order to run the game, go ahead and execute the Python code “Happy Garden”. It will start off with an image of a happy animal in the grass field with flowers placed randomly on the game board. The goal is to water the flowers without getting hit by the angry flowers. You will be prompted with a timer to see how long you can last. In order to move the character, use the keyboard arrows and the space bar to water the flowers.

To run Hello World to OpenAI, we need to download Python but in a version between 3.7 and 3.10. The openai-quickstart-python-master file is downloaded and powershell is opened while on administrator. The path of PowerShell is set to the folder. The .env file is created using PowerShell with cd open-quickstart-python and the contents of the .env. an example is copied to the new .env file. The API key is placed in the .env file. A virtual environment is then created and activated using python -m venv venv and .\venv\Scripts\activate.bat respectively. Pip install -r requirements.txt is done and the flask run command is done to run the flask. We are now connected and <http://localhost:5000> allows us to access the website and OpenAI content using our API key.

To run Hello World to ChatGPT, the ChatGPT\_LoopQuery.py file is ran and the API key is placed on line 9. The program can now be run.

# DESIGN ARCHITECTURE

1. Install necessary Python packages

pip install tensorflow

pip install keras

pip install h5py

pip install Matplotlib

pip install numpy

# PROCESS & WORKFLOW

Part 1: perform the steps outlined https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/images/cnn.ipynb#scrollTo=WRzW5xSDDbNF

This baseline code achieved a test accuracy of over 70%.

-> Follow/open the file: cnn.py

-> You can see that we are increasing the accuracy from 70% to 87%

-> Line 40 Create the convolutional base, which adds more to the model

->Line 62 Add Dense layers on top

-> Line 70 Compile and train the model

Part 2: example from the lecture open CNNbaseline.py to learn AND open the test\_impage.py to test the images

open the text.py that we created to increase the accuracy from 70% to 90%+

Part 3: Game Development: Happy Garden

Open the Happy\_Garden.py code and execute the Python code. This will open the GUI for the game, and you will see the farm animal in the grass field with flowers surrounding it. You will move the animal around and water the flowers. Move around the animal using the arrow keys and press the space bar to water the flowers.

Part 4: Hello World to OpenAi

Open the HelloWorld.py code and execute the Python code. This will open the GUI for the openAI code. The AI will prompt you at least four names for any animal that you ask. For example, you can say cat, and four names will be displayed that are clever names. To add your own API, go to the .env file and replace the line with your API. To change the picture, go to the main folder, look for the template, and insert a PNG. To change the title, go to the index.html and change line 13. To change the amount of names generated, you can go to app.py to change to generate four.

Part 5: Hello World to ChatGPT

Open the ChatGPT.py code and execute the Python code. This will open the GUI to the ChatGPT code. To change the engine, go to line 20 on mode\_engine to change the engine. The list of engines are located on lines 48 through 52. To change the prompt, go to line 36 to change the prompt on the input line. To change the amount of tokens, go to line 26 to change the value of max tokens. Line 26 changes the value of n. Line 28 changes what other input is needed to stop it. Line 29 changes the temperature.

# TEST DATA

Part 1: CNN baseline

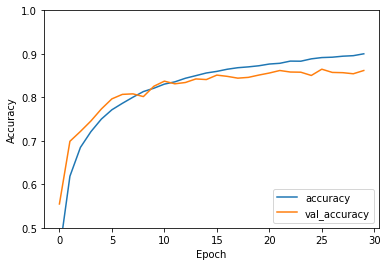




Figure 1: CNN baseline.

Part 2: CNN Challenge Test

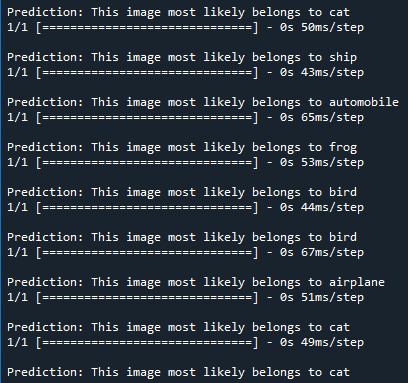


Figure 2: CNN Challenge Test.

Part 3: Game Development

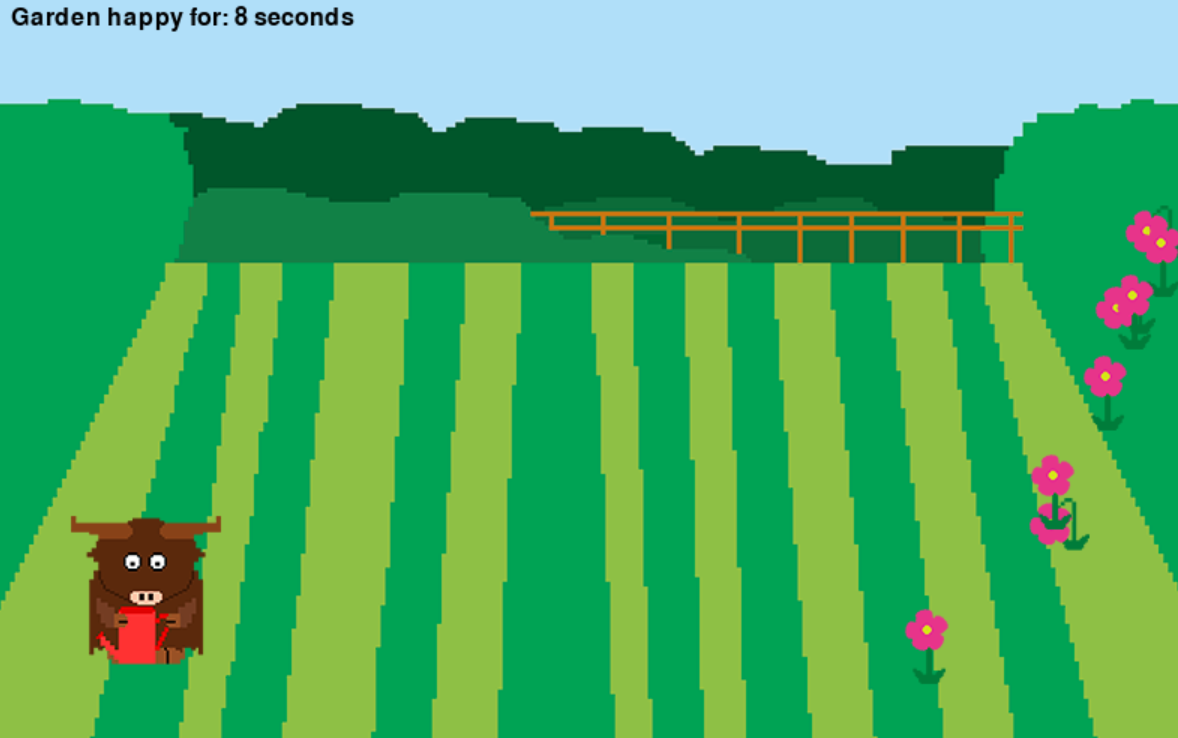


Figure 3: Game Development Garden Happy. bnm

Part 4: Hello World to OpenAi



Figure 4: Baseline Hello World to OpenAI Figure 5: Picture change Hello World to OpenAI



Figure 6: Four generations Hello World to OpenAI Figure 7: Change title Hello World to OpenAI

Part 5: Hello World to ChatGPT

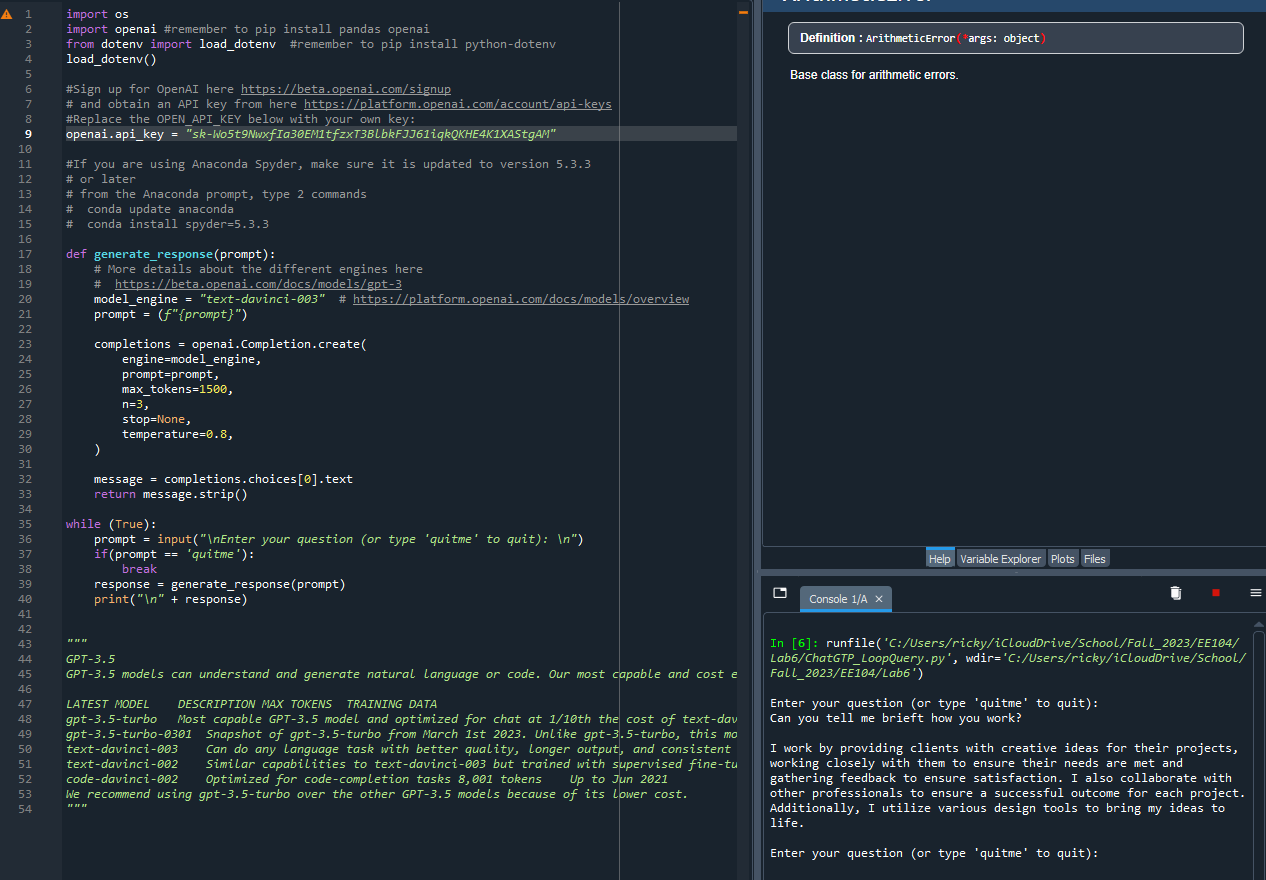


Figure 8: Hello World to ChatGPT baseline code.

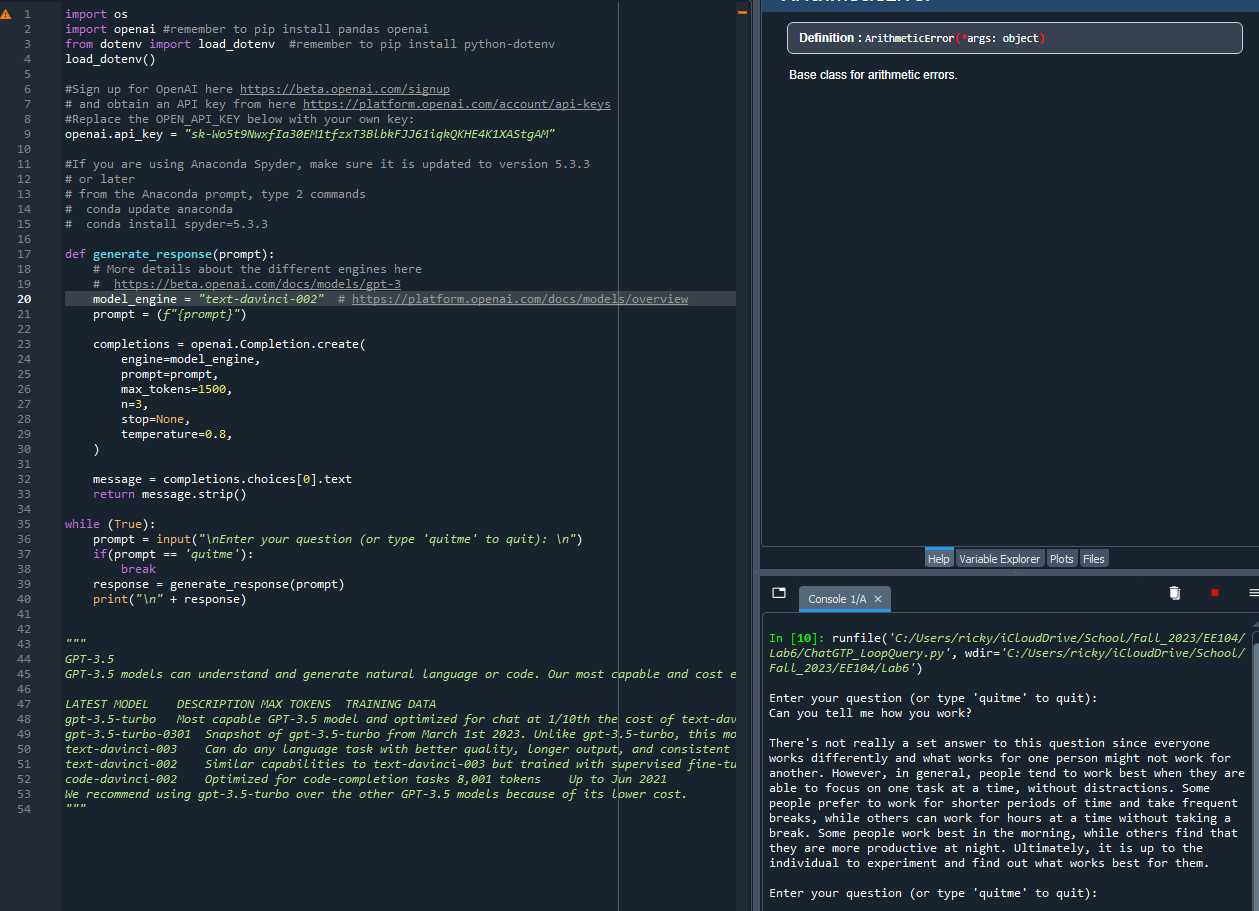


Figure 9: Hello World to ChatGPT engine change.

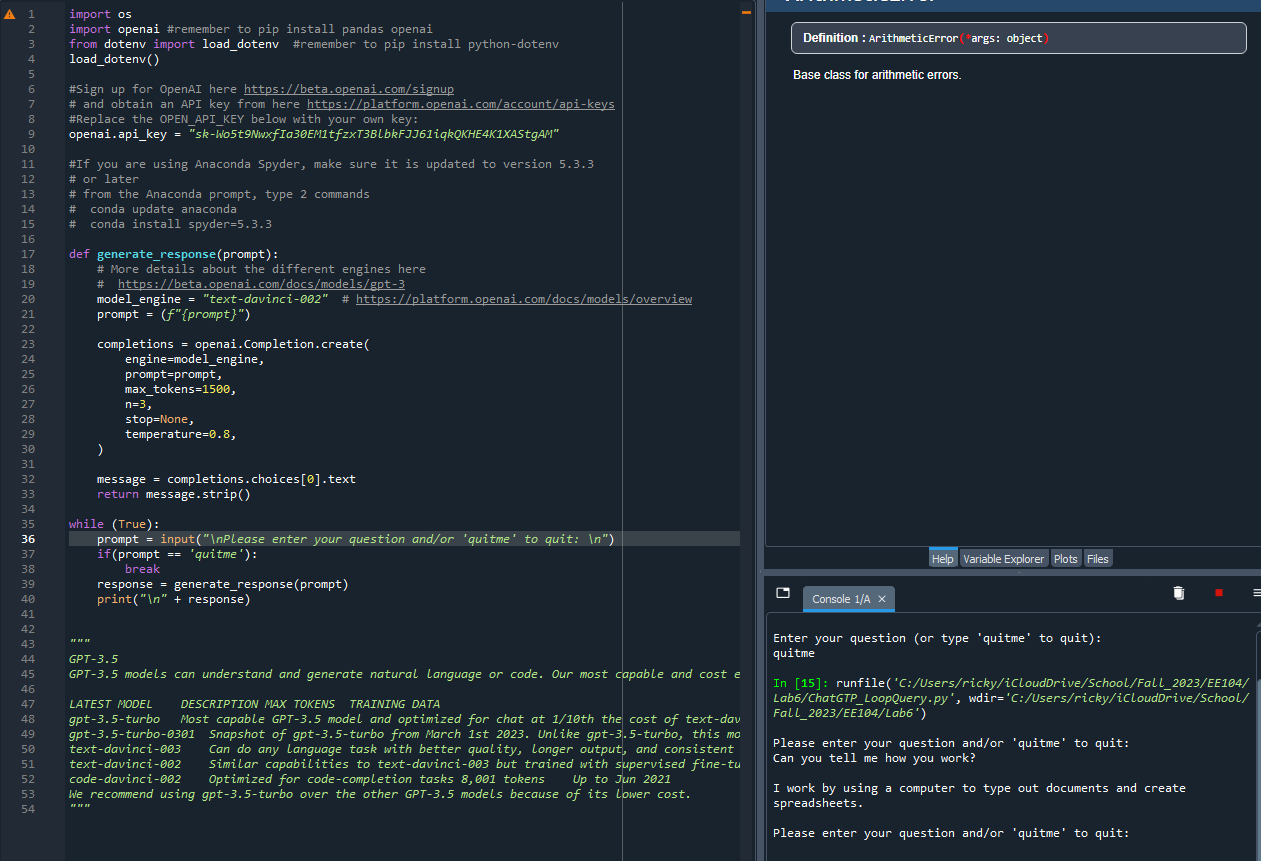


Figure 10: Hello World to ChatGPT prompt change.

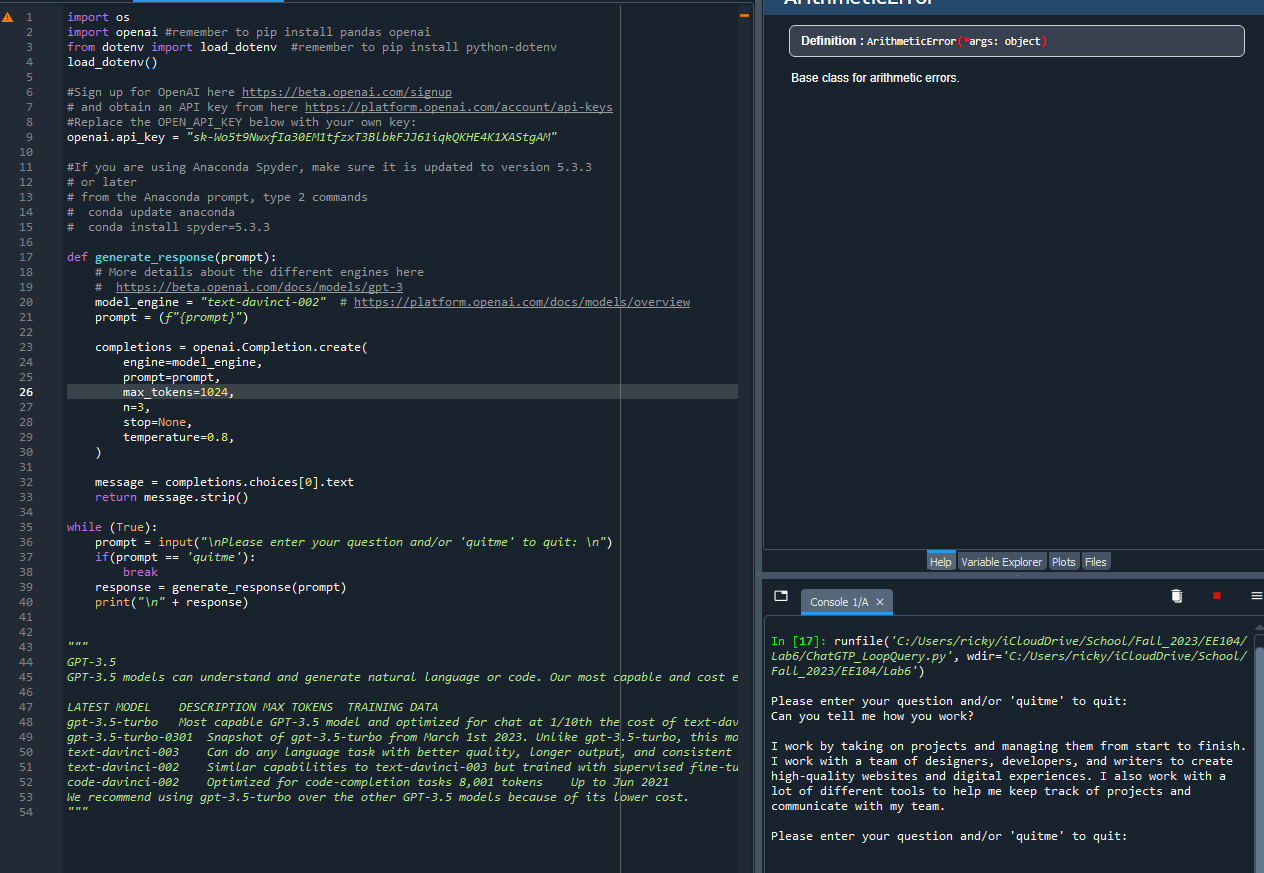


Figure 11: Hello World to ChatGPT max token change.

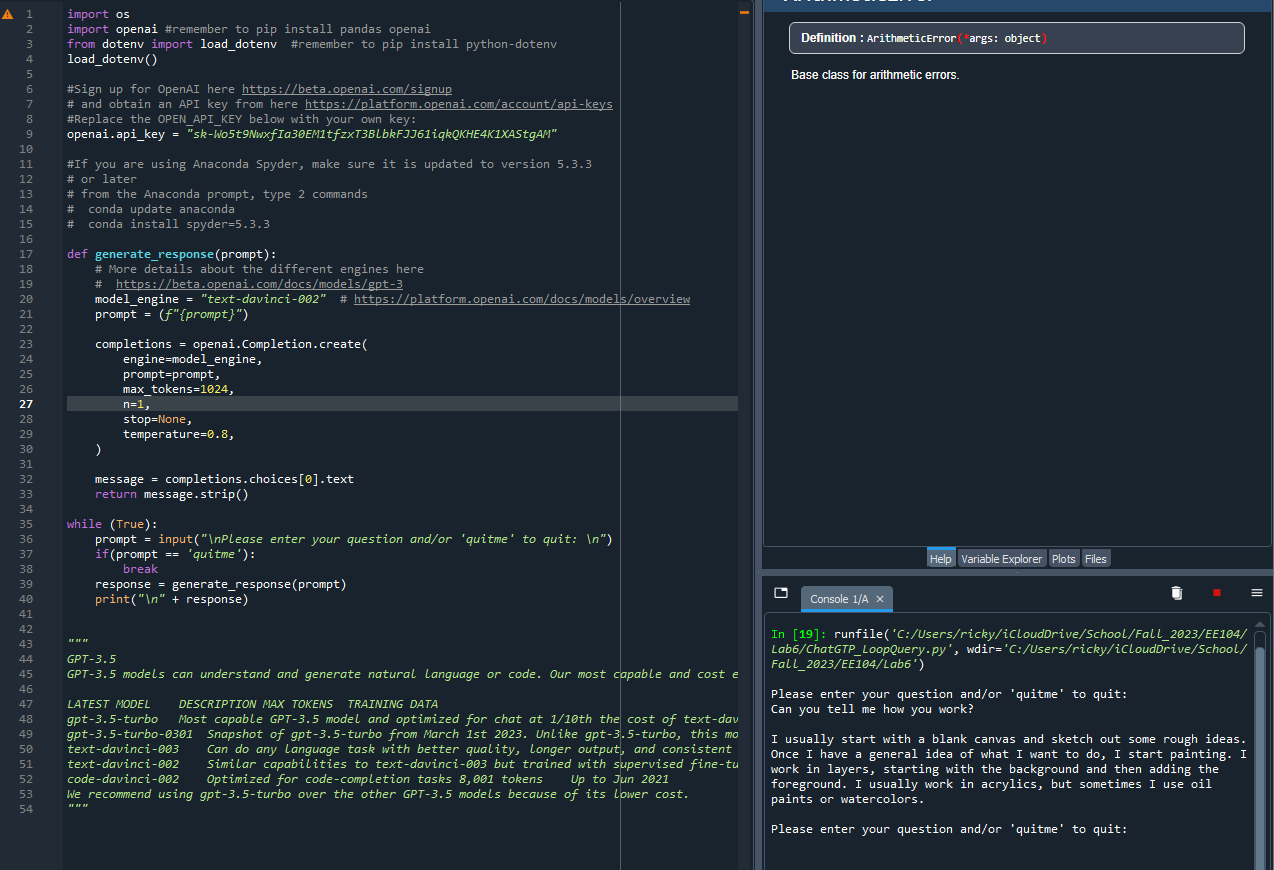


Figure 12: Hello World to ChatGPT n change.

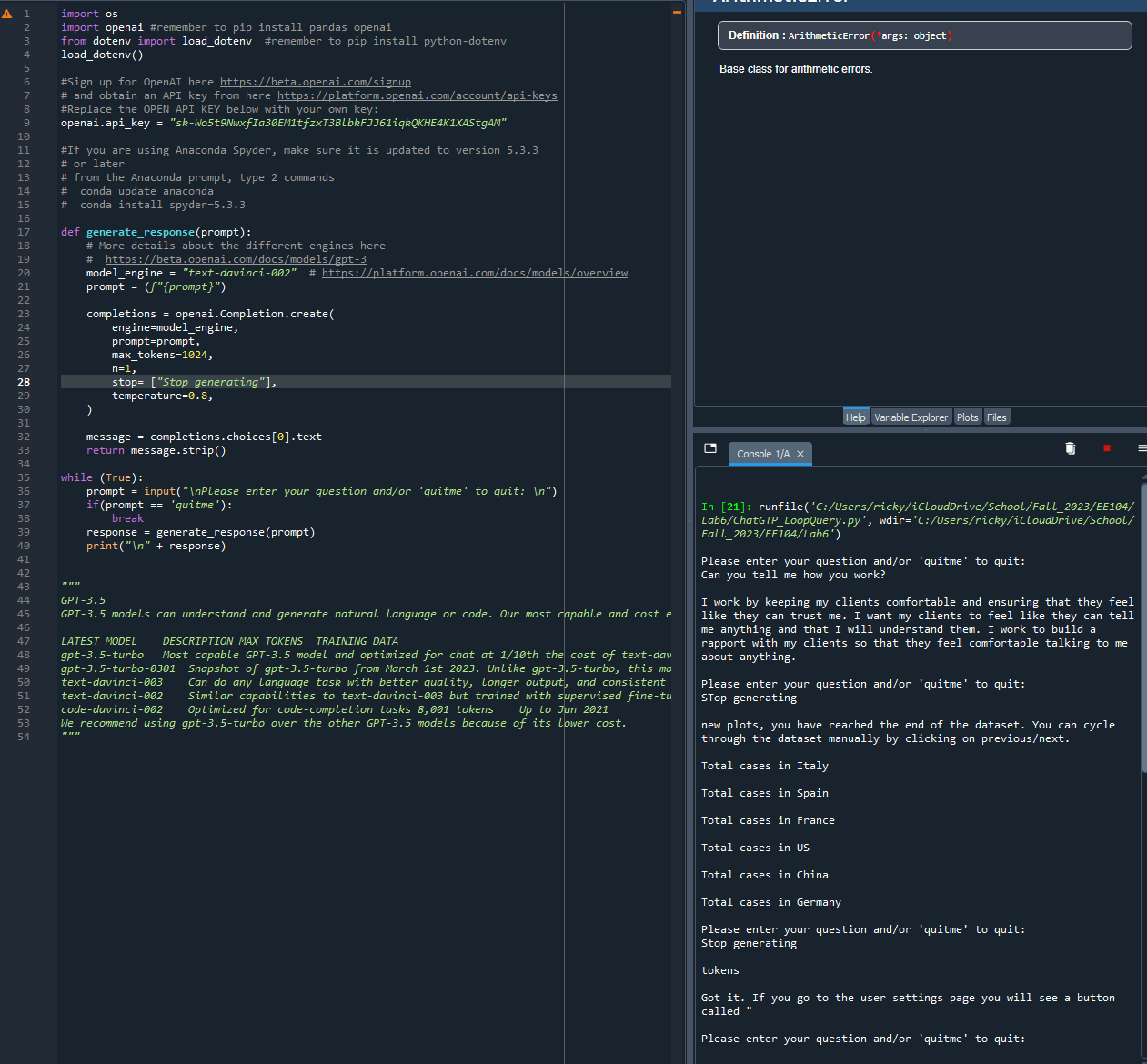


Figure 5: Hello World to ChatGPT stop change.

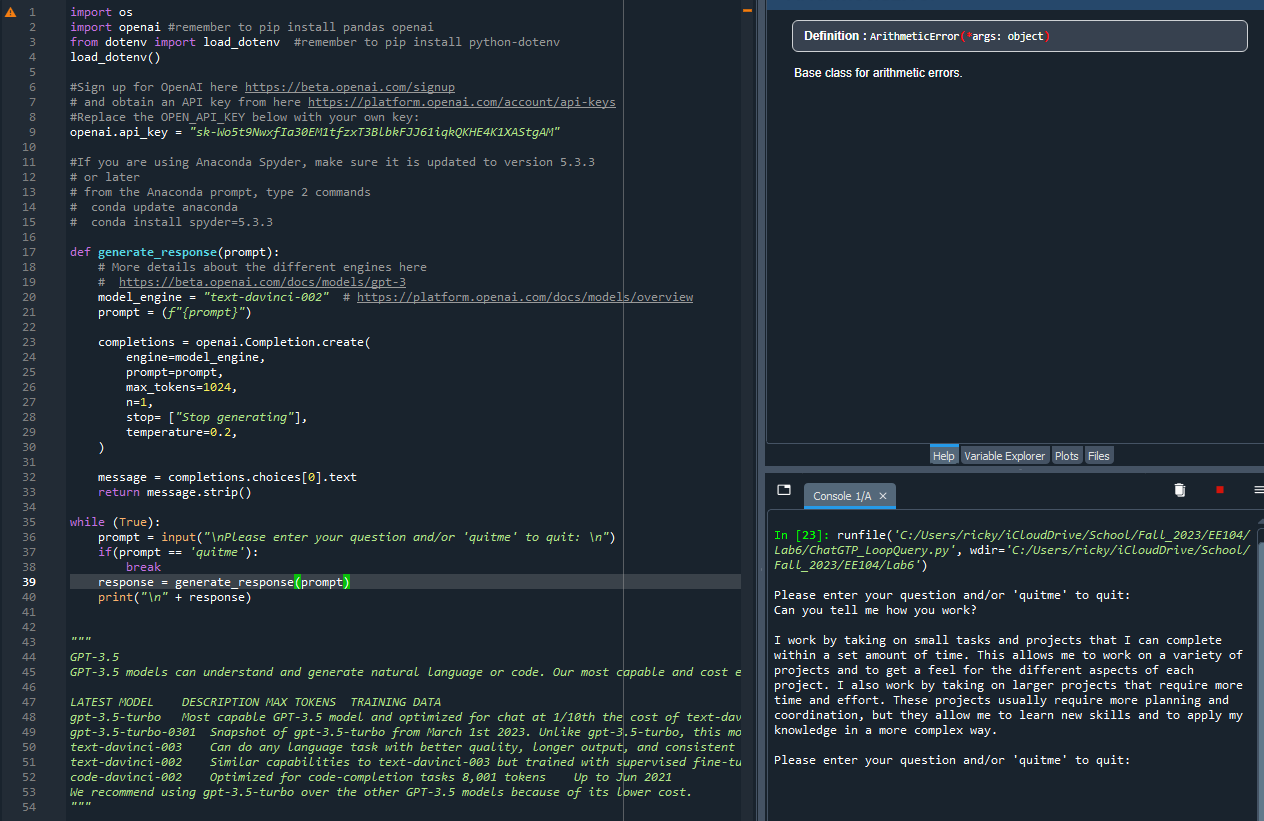


Figure 5: Hello World to ChatGPT temperature change.

# VIDEO RECORDINGS

| **Recording Title** | **URL** | **Notes** |
| --- | --- | --- |
| CNN - Baseline + Increasing Dropout + Data Augmentation + Batch Normalization + Your own method | https://youtu.be/ZoqjOlLAyvM | CNN baseline with added methods such as baseline, increasing dropout, data augmentation, and batch normalization. |
| CNN - Challenge test | https://youtu.be/cuRngM0nEt4 | Testing images and the prediction, this will be loaded from the CNN baseline file. |
| Game Development | https://youtu.be/NTmeIgpSG1w |  |
| Hello World to OpenAI | https://youtu.be/fwMy9gkx6UU | Hello World to OpenAI Gemini |
| Hello World to ChatGTP | https://youtu.be/Ividksc0ifs | Hello World to ChatGTP Gemini |
| GitHub |  | GitHib URL |

# CONCLUSIONS

This lab consisted of a wide array of AI-related tasks, allowing us to develop our skills and understanding this dynamic field. the lab started with trying to fine-tune a Convolutional Neural Network, pushing its accuracy from 70% to 90%, while also making sure it can effectively identify 90% of the test images, thanks to techniques like dropout, data augmentation, and batch normalization. We were able to successfully get the accuracy of the code to about 87%. From there, we used the image\_test.py code as an example so that we could recognize unrecognizable images and test the limits of our neural network. For the third part of the lab, we worked on the Happy Garden game, getting hands-on experience with game modification, tweaks, and enhancements. The realm of OpenAI brought its own set of opportunities as we harnessed Python to bring about changes on a website, from image and title alterations to tweaking code to generate more names than before. The last part of the lab consisted of fine-tuning a ChatGPT model, providing us with a practical understanding of how AI language models work. This comprehensive experience has not only broadened our knowledge but also equipped us with practical skills in artificial intelligence, image recognition, game development, and natural language processing. It's exciting to see the application of these techniques in various domains, and this journey has undoubtedly laid a strong foundation for our academic and professional growth.

# REFERENCES

[1]. Brownlee, J. (2019, May 12). *How to Develop a CNN From Scratch for CIFAR-10 Photo Classification*. Machine Learning Mastery. <https://machinelearningmastery.com/how-to-develop-a-cnn-from-scratch-for-cifar-10-photo-classification/>

*[2]. Google Colaboratory*. (n.d.). Colab.research.google.com. Retrieved November 5, 2023, from https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/images/cnn.ipynb#scrollTo=WRzW5xSDDbNF

[3]. Krizhevsky, A. (2009). *CIFAR-10 and CIFAR-100 datasets*. Toronto.edu. <https://www.cs.toronto.edu/~kriz/cifar.html>

[4]. Winner, L. (1980). *Do artifacts have politics?* Daedalus.