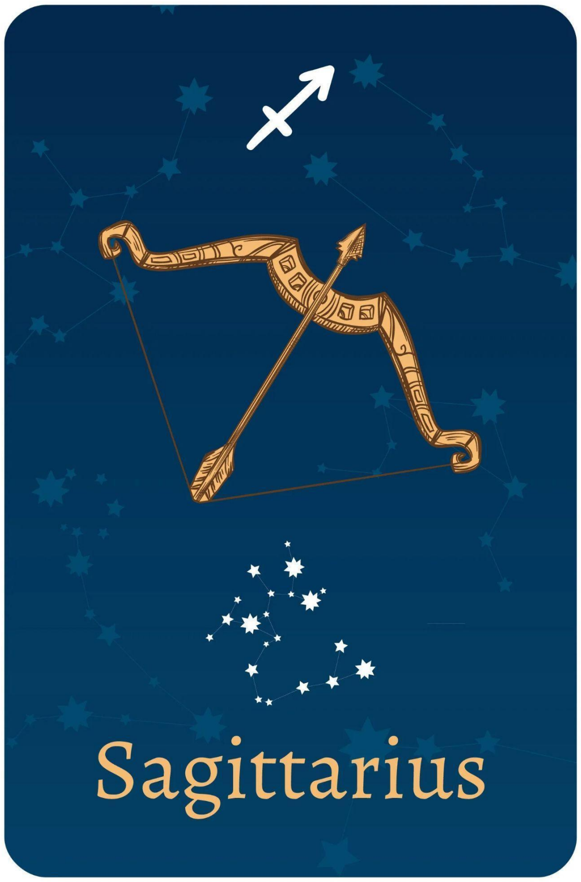
SJSU SAN JOSÉ STATE UNIVERSITY

Department of Electrical Engineering
EE104 - Fall 2023



TEAM MEMBERS

Shivansh Shukla

Nathan Lee

Edwin Kulakkattolikel

Contents

PURPOSES AND GOALS

Recognize CNNs:

Understand the foundations of CNN image recognition.

Learn about CNN architecture and image categorization operations.

Dataset CIFAR-10:

Familiarize yourself with the CIFAR-10 dataset, which has ten image classes.

Master machine learning data loading and preprocessing.

Code de base:

Implement the tutorial's baseline CNN code.

Attain a test accuracy of more than 90%.

Improve Accuracy:

To increase model performance, modify the baseline code.

Use lecture notes and resources to improve your techniques.

Image Customization:

Use custom photos to train the CNN model.

GitHub and the Lab Report:

Create a lab report and upload your code to GitHub.

Performance Objectives:

Attain the required accuracy for various point prizes.

Challenge Exam:

Recognize three difficult images for points.

Game Creation:

Point-earning hacks and changes can be applied to Big Quiz project.

Hello World to OpenAI and ChatGPT:

Execute OpenAI tasks with point-earning customizations.

HOW TO INSTALL THE PROGRAMS

DEPENDENCIES

CNN:

pip install matplotlib

pip install keras

pip install tensorflow

An internet connection may be necessary for downloading the CIFAR-10 dataset if it's not already available on your system**.**
Storage: The code saves the trained model as 'NathanL_CIFARmodel_baseline.h5', so ensure you have write permissions in the working directory.

Test Image:

pip install tensorflow

pip install matplotlib

pip install certifi
Internet Connection: The code fetches images from the internet using their URLs. You must have an internet connection to access and download these images. The code uses the get_file method from TensorFlow to download the images.
HTTPS Verification: The code includes logic to disable SSL certificate verification. This is necessary when loading a model from a remote server. It's handled via the PYTHONHTTPSVERIFY environment variable.

Baseline Model: Trained baseline model needed from previous saved from CNN code 'NathanL_CIFARmodel_baseline.h5'

Big Quiz Game:

pip install pygame

pip install pgzrun

Hello World to OpenAI:

To install and run the provided Python Flask application, you'll need to ensure that your system has Python installed. If not, you can download and install Python from the official Python website. Additionally, confirm that Pip, the Python package manager, is available and up to date.

It's advisable to create a virtual environment to manage the project's dependencies. You can do this by using the python -m venv venv command on Windows or python3 -m venv venv on macOS and Linux. Once created, activate the virtual environment.

The Flask application relies on various Python packages, which can be installed using Pip. Navigate to the project directory where the provided code is located and install these dependencies listed in a requirements.txt file (if provided) by running pip install -r requirements.txt.

The Flask application expects the OpenAI API key to be set as an environment variable named "OPENAI_API_KEY." Ensure you have an OpenAI API key and set it as an environment variable.

Access the application through a modern web browser, such as Google Chrome, Mozilla Firefox, or Microsoft Edge. The provided code is platform-independent, so it should work on different operating systems, including Windows 11, macOS, and various Linux distributions.

Once you've met these requirements, execute the Python script app.py from the project directory. Then, open your web browser and navigate to the application's URL, typically http://localhost:5000. This allows you to interact with the web page, enter an animal name, and generate superhero names.

Hello World to ChatGPT:

You should start by ensuring you have Python installed on your computer. It's essential to have Python 3.6 or a later version. Following that, use the Python package manager, pip, to install the required packages. Open your command prompt or terminal and execute the following commands: pip install openai and pip install python-dotenv. The "openai" package facilitates interaction with the OpenAI API, while "python-dotenv" is employed to load environment variables from a .env file. Prior to running the code, create an account with OpenAI and obtain an API key from their platform. Store this API key in a .env file located in the project directory like so: OPENAI_API_KEY=your_api_key_here.

Ensure that "your_api_key_here" is replaced with your specific API key. You can utilize any text editor or integrated development environment (IDE) compatible with Python to execute the code. If you're an Anaconda Spyder user, ensure that Spyder is updated to version 5.3.3 or a later release using the provided commands. Once you've completed these setup steps, execute the code, which operates as a basic chatbot using the OpenAI API, repeatedly requesting user input until "quitme" is entered to exit the loop.

HOW TO RUN THE PROGRAMS

CNN: Run code. Wait 7 hours. See final validation accuracy

Test Image: Choose images to test. Create/find link for images. Run Code. Check validation of classification.

Big Quiz Game:

In order to run the game, the user needs to have Python installed with a preferred IDE along with the libraries pygame and pgzrun. With these installed, the user can use the provided code and use the kernel that sends alerts to the user like getting hints for the questions.

Hello World to OpenAI:

To run the Flask application, ensure you have Python and Pip installed. Activate a virtual environment if you've created one. Install the required dependencies from a requirements.txt file. Set the "OPENAI_API_KEY" environment variable. Run the application with python app.py. Access it in your web browser at http://localhost:5000, where you can input an animal name to generate superhero names based on your input.

Hello World to ChatGPT:

Begin by opening your computer's terminal or command prompt, typically accessible through the system's applications or search feature. Next, navigate to the directory where you've saved the Python script containing the chatbot code by using the cd command, specifying the path to your program directory. If you're using a virtual environment, activate it according to your platform's commands. Then, execute the chatbot program by running the Python script, replacing "your_script.py" with the actual script filename. Once the script is running, interact with the chatbot by entering questions or statements in response to its prompts. To exit the chatbot and terminate the program, simply type "quitme" and press Enter. These steps enable you to easily run the program and engage with the chatbot, receiving responses generated by the OpenAI GPT-3.5 model.

DESIGN ARCHITECTURE

CNN: Hardware Components are the CPU, which performs the computations necessary for training and evaluating the neural network, and the memory which stores the baseline data. Software Logical Blocks are the Spyder IDE, Python Code, CIFAR-10 Dataset, baseline model, and libraries such as Keras, Tensorflow, and Matplotlib. This code uses TensorFlow with Keras for deep learning, loading and processing the CIFAR-10 dataset. It augments the data, builds a CNN model, visualizes results with Matplotlib, and evaluates the model's performance.

Test Image: This code relies on a computer with internet access and various software components, including Python, TensorFlow, and image processing libraries. It loads a pre-trained model and processes images downloaded from the internet, displaying the images and their predicted class labels. The core logic is centered around image classification using the pre-trained model.

Big Quiz Game: For this program to run, it requires hardware components such as the CPU and the computer system, as well as the peripherals such as the display and input devices (keyboard and mouse). It also requires an operating system that is capable of running python script using a preferred IDE, in which this code accesses the imported libraries.

Hello World to OpenAI:

The front-end web interface, which includes HTML and CSS serves as the user-facing part of the application, responsible for rendering the web page and handling user input. The Flask web application, a local software component, acts as the server-side logic. It processes HTTP requests, communicates with the OpenAI GPT-3 API, and can potentially interact with a local database for data storage.

The OpenAI GPT-3 API, a cloud-based service, is a critical component responsible for text generation based on user input. It relies on powerful cloud-based hardware and machine learning models to generate superhero names. Additionally, a web server, whether hosted locally for development or on a cloud-based service for production, provides the infrastructure for serving the Flask application and making it accessible via the internet. Lastly, the end-user's web browser is the client-side component that interacts with the web interface, allowing users to input animal names and receive generated superhero names. This architecture combines both local and cloud-based components to deliver the desired functionality to users, with each logical block playing a specific role in the overall system.

Hello World to ChatGPT: The code architecture involves a symbiotic relationship between hardware and software components. Hardware blocks form the physical computing infrastructure, while software components, including Python and related packages, manage user interactions. At its core, the OpenAI cloud-based GPT-3.5 model plays a pivotal role in processing user input and producing responses. This interplay between hardware and software enables fluid communication between the user and the chatbot, with data flowing between local and cloud-based resources, allowing for dynamic conversation and interaction.

PROCESS & WORKFLOW

CNN: This code loads the CIFAR-10 dataset, preprocesses the data, defines and trains a CNN model with data augmentation, evaluates the model's accuracy, and saves the trained model.

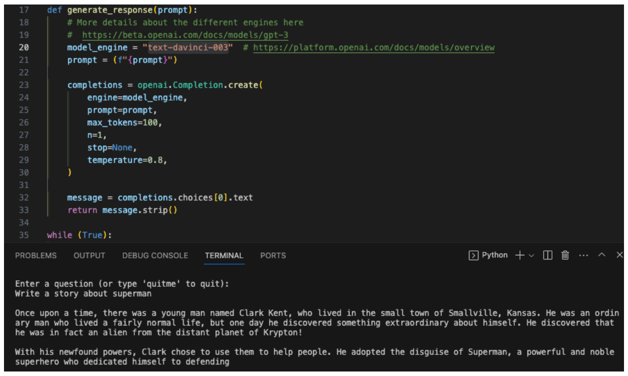
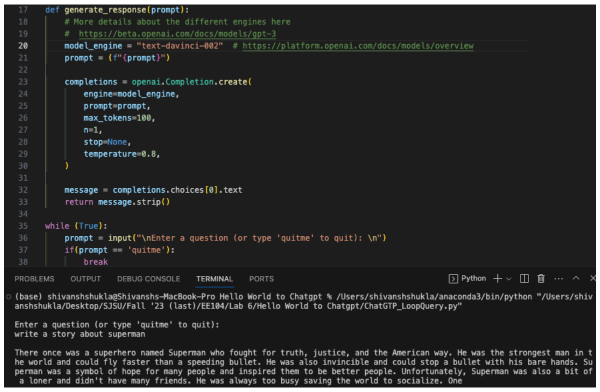
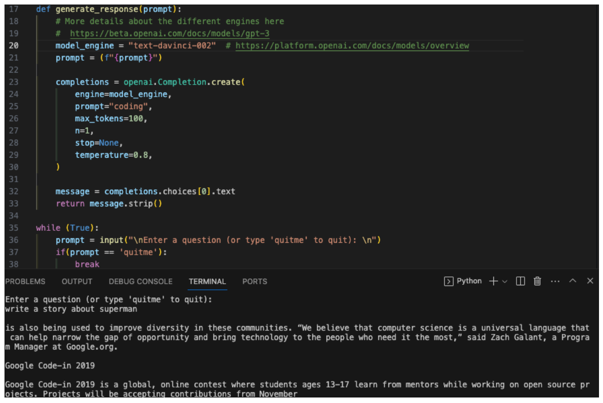
Test Image: This code loads the pre-trained model from the CNN code and pulls images from the internet. The code then runs the image through the classification process based on the CNN model. The code then classifies the images.

Big Quiz Game: In this program, the game goes through the python code while accessing the provided libraries. Once the program is ran, the user is then provided with multiple choice questions along with a 15 second timer. When the user selects the correct answer, the user is then provided with the next question in the list and the timer is then reset to 15 seconds. If the user does not answer the question within the time limit, the user is provided with the dialogue “Game over” along with their score. The “game over” screen is also displayed if a user selects the wrong answer. If the user encounters a question that they are unable to answer, the user can press the “H” key on the keyboard, which will then prompt the hint in the kernel in the IDE.

Hello World to OpenAI: The process begins with the user's interaction with the web application, where they input an animal's name. The Flask server receives this input and forwards it to the OpenAI GPT-3 API. The API generates superhero names based on the provided animal name and returns them to the server. Subsequently, the web browser displays the generated superhero names to the user, concluding the process. In terms of workflow, the user starts by accessing the web application, where they input an animal name. The server processes the input and collaborates with the API for text generation. The user receives the results on the web page, completing their interaction with the application.

Hello World to ChatGPT: The chatbot process begins as the user inputs questions or statements, which are transmitted to the cloud-based OpenAI model for response generation. This interaction loop continues until the user opts to exit by typing "quitme." In the workflow, the user initiates interaction by running the code, entering queries, and receiving responses. This process can be iterative, with the user asking multiple questions, and the chatbot responds accordingly. At any point, the user can choose to conclude the interaction by typing "quitme," which leads to program termination and the user's exit. This structured workflow ensures a user-friendly and efficient exchange of information in a conversational manner.

TEST DATA

Test Name	Output
Hello World to ChatGPT: Engine	 
Hello World to ChatGPT: prompt	

```python
def generate_response(prompt):
    # More details about the different engines here
    #  https://beta.openai.com/docs/models/gpt-3
    model_engine = "text-davinci-003"  # https://platform.openai.com/docs/models/overview
    prompt = (f"{prompt}")

    completions = openai.Completion.create(
        engine=model_engine,
        prompt=prompt,
        max_tokens=100,
        n=1,
        stop=None,
        temperature=0.8,
    )

    message = completions.choices[0].text
    return message.strip()

while (True):
```

```
Enter a question (or type 'quitme' to quit):
Write a story about superman

Once upon a time, there was a young man named Clark Kent, who lived in the small town of Smallville, Kansas. He was an ordin
ary man who lived a fairly normal life, but one day he discovered something extraordinary about himself. He discovered that
he was in fact an alien from the distant planet of Krypton!

With his newfound powers, Clark chose to use them to help people. He adopted the disguise of Superman, a powerful and noble
superhero who dedicated himself to defending
```

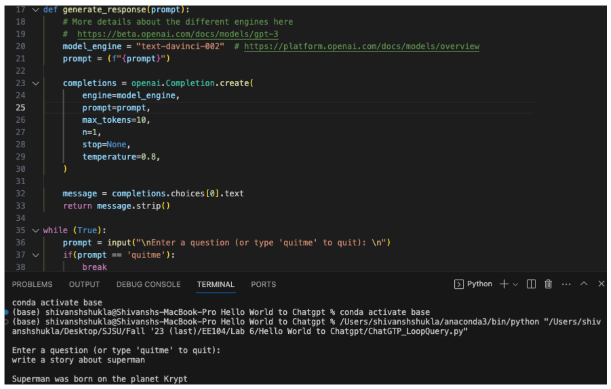
```python
def generate_response(prompt):
    # More details about the different engines here
    #  https://beta.openai.com/docs/models/gpt-3
    model_engine = "text-davinci-002"  # https://platform.openai.com/docs/models/overview
    prompt = (f"{prompt}")

    completions = openai.Completion.create(
        engine=model_engine,
        prompt=prompt,
        max_tokens=100,
        n=1,
        stop=None,
        temperature=0.8,
    )

    message = completions.choices[0].text
    return message.strip()

while (True):
    prompt = input("\nEnter a question (or type 'quitme' to quit): \n")
    if(prompt == 'quitme'):
        break
```

```
(base) shivanshshukla@Shivanshs-MacBook-Pro Hello World to Chatgpt % /Users/shivanshshukla/anaconda3/bin/python "/Users/shiv
anshshukla/Desktop/SJSU/Fall '23 (last)/EE104/Lab 6/Hello World to Chatgpt/ChatGTP_LoopQuery.py"

Enter a question (or type 'quitme' to quit):
write a story about superman

There once was a superhero named Superman who fought for truth, justice, and the American way. He was the strongest man in t
he world and could fly faster than a speeding bullet. He was also invincible and could stop a bullet with his bare hands. Su
perman was a symbol of hope for many people and inspired them to be better people. Unfortunately, Superman was also a bit of
 a loner and didn't have many friends. He was always too busy saving the world to socialize. One
```

```python
def generate_response(prompt):
    # More details about the different engines here
    #  https://beta.openai.com/docs/models/gpt-3
    model_engine = "text-davinci-002"  # https://platform.openai.com/docs/models/overview
    prompt = (f"{prompt}")

    completions = openai.Completion.create(
        engine=model_engine,
        prompt="coding",
        max_tokens=100,
        n=1,
        stop=None,
        temperature=0.8,
    )

    message = completions.choices[0].text
    return message.strip()

while (True):
    prompt = input("\nEnter a question (or type 'quitme' to quit): \n")
    if(prompt == 'quitme'):
        break
```

```
Enter a question (or type 'quitme' to quit):
write a story about superman

is also being used to improve diversity in these communities. “We believe that computer science is a universal language that
 can help narrow the gap of opportunity and bring technology to the people who need it the most,” said Zach Galant, a Progra
m Manager at Google.org.

Google Code-in 2019

Google Code-in 2019 is a global, online contest where students ages 13-17 learn from mentors while working on open source pr
ojects. Projects will be accepting contributions from November
```

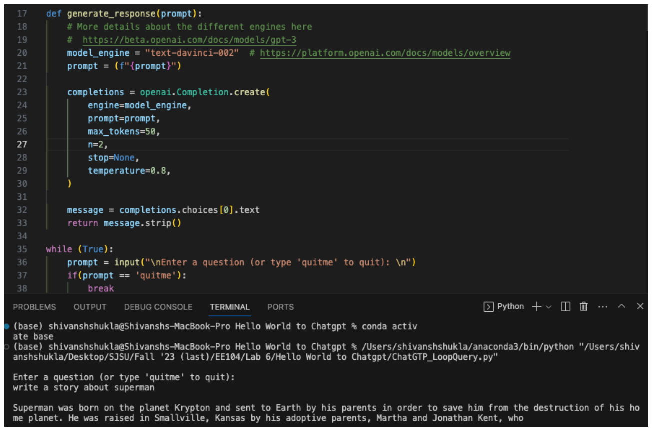
Hello World to ChatGPT: max tokens



```python
def generate_response(prompt):
    # More details about the different engines here
    #  https://beta.openai.com/docs/models/gpt-3
    model_engine = "text-davinci-002"  # https://platform.openai.com/docs/models/overview
    prompt = (f"{prompt}")

    completions = openai.Completion.create(
        engine=model_engine,
        prompt=prompt,
        max_tokens=10,
        n=1,
        stop=None,
        temperature=0.8,
    )

    message = completions.choices[0].text
    return message.strip()

while (True):
    prompt = input("\nEnter a question (or type 'quitme' to quit): \n")
    if(prompt == 'quitme'):
        break
```

```
conda activate base
(base) shivanshshukla@Shivanshs-MacBook-Pro Hello World to Chatgpt % conda activate base
(base) shivanshshukla@Shivanshs-MacBook-Pro Hello World to Chatgpt % /Users/shivanshshukla/anaconda3/bin/python "/Users/shiv
anshshukla/Desktop/SJSU/Fall '23 (last)/EE104/Lab 6/Hello World to Chatgpt/ChatGTP_LoopQuery.py"

Enter a question (or type 'quitme' to quit):
write a story about superman

Superman was born on the planet Krypt
```

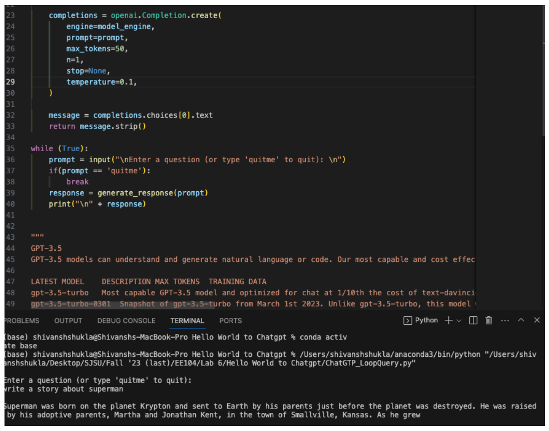
Hello World to ChatGPT: n



```python
def generate_response(prompt):
    # More details about the different engines here
    #  https://beta.openai.com/docs/models/gpt-3
    model_engine = "text-davinci-002"  # https://platform.openai.com/docs/models/overview
    prompt = (f"{prompt}")

    completions = openai.Completion.create(
        engine=model_engine,
        prompt=prompt,
        max_tokens=50,
        n=2,
        stop=None,
        temperature=0.8,
    )

    message = completions.choices[0].text
    return message.strip()

while (True):
    prompt = input("\nEnter a question (or type 'quitme' to quit): \n")
    if(prompt == 'quitme'):
        break
```

```
(base) shivanshshukla@Shivanshs-MacBook-Pro Hello World to Chatgpt % conda activ
ate base
(base) shivanshshukla@Shivanshs-MacBook-Pro Hello World to Chatgpt % /Users/shivanshshukla/anaconda3/bin/python "/Users/shiv
anshshukla/Desktop/SJSU/Fall '23 (last)/EE104/Lab 6/Hello World to Chatgpt/ChatGTP_LoopQuery.py"

Enter a question (or type 'quitme' to quit):
write a story about superman

Superman was born on the planet Krypton and sent to Earth by his parents in order to save him from the destruction of his ho
me planet. He was raised in Smallville, Kansas by his adoptive parents, Martha and Jonathan Kent, who
```

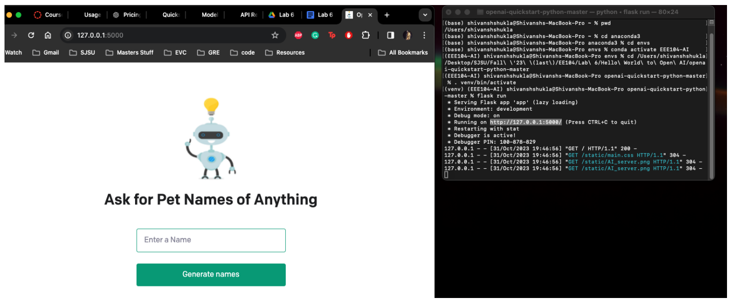
Hello World to ChatGPT: temperature



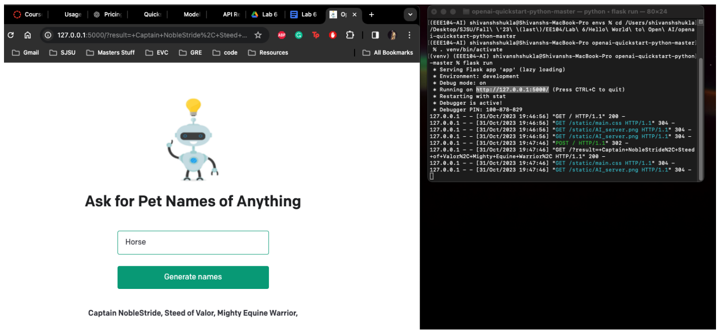
```python
    completions = openai.Completion.create(
        engine=model_engine,
        prompt=prompt,
        max_tokens=50,
        n=1,
        stop=None,
        temperature=0.1,
    )

    message = completions.choices[0].text
    return message.strip()

while (True):
    prompt = input("\nEnter a question (or type 'quitme' to quit): \n")
    if(prompt == 'quitme'):
        break
    response = generate_response(prompt)
    print("\n" + response)


"""
GPT-3.5
GPT-3.5 models can understand and generate natural language or code. Our most capable and cost effec

LATEST MODEL    DESCRIPTION MAX TOKENS  TRAINING DATA
gpt-3.5-turbo   Most capable GPT-3.5 model and optimized for chat at 1/10th the cost of text-davinci
gpt-3.5-turbo-0301  Snapshot of gpt-3.5-turbo from March 1st 2023. Unlike gpt-3.5-turbo, this model
```

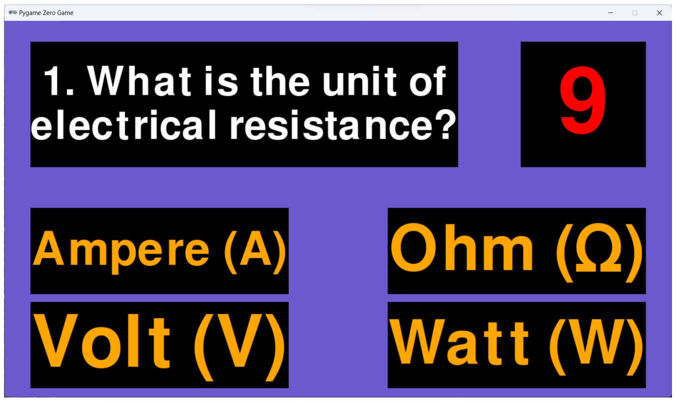
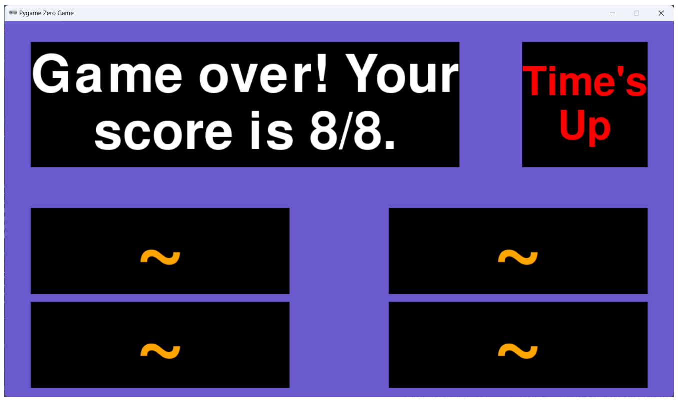
```
(base) shivanshshukla@Shivanshs-MacBook-Pro Hello World to Chatgpt % conda activ
ate base
(base) shivanshshukla@Shivanshs-MacBook-Pro Hello World to Chatgpt % /Users/shivanshshukla/anaconda3/bin/python "/Users/shiv
anshshukla/Desktop/SJSU/Fall '23 (last)/EE104/Lab 6/Hello World to Chatgpt/ChatGTP_LoopQuery.py"

Enter a question (or type 'quitme' to quit):
write a story about superman

Superman was born on the planet Krypton and sent to Earth by his parents just before the planet was destroyed. He was raised
 by his adoptive parents, Martha and Jonathan Kent, in the town of Smallville, Kansas. As he grew
```

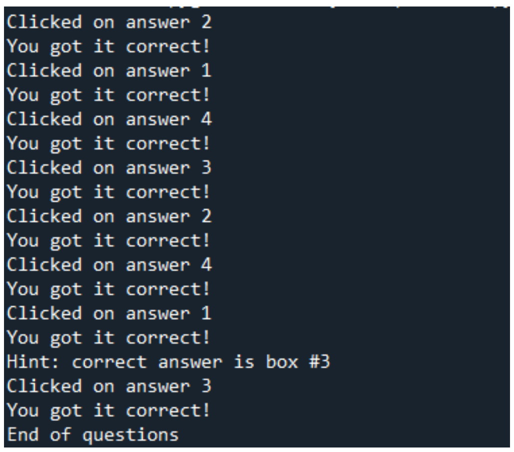
Hello World to OpenAI



```
openai-quickstart-python-master — python ‹ flask run — 80×24
(base) shivanshshukla@Shivanshs-MacBook-Pro ~ % pwd
/Users/shivanshshukla
(base) shivanshshukla@Shivanshs-MacBook-Pro ~ % cd anaconda3
(base) shivanshshukla@Shivanshs-MacBook-Pro anaconda3 % cd envs
(base) shivanshshukla@Shivanshs-MacBook-Pro envs % conda activate EEE104-AI
(EEE104-AI) shivanshshukla@Shivanshs-MacBook-Pro envs % cd /Users/shivanshshukla
/Desktop/SJSU/Fall\ \'23\ \(last\)/EE104/Lab\ 6/Hello\ World\ to\ Open\ AI/opena
i-quickstart-python-master
(EEE104-AI) shivanshshukla@Shivanshs-MacBook-Pro openai-quickstart-python-master
 % . venv/bin/activate
(venv) (EEE104-AI) shivanshshukla@Shivanshs-MacBook-Pro openai-quickstart-python
-master % flask run
 * Serving Flask app 'app' (lazy loading)
 * Environment: development
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 100-878-829
127.0.0.1 - - [31/Oct/2023 19:46:56] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [31/Oct/2023 19:46:56] "GET /static/main.css HTTP/1.1" 304 -
127.0.0.1 - - [31/Oct/2023 19:46:56] "GET /static/AI_server.png HTTP/1.1" 304 -
127.0.0.1 - - [31/Oct/2023 19:46:56] "GET /static/AI_server.png HTTP/1.1" 304 -
```

	 Ask for Pet Names of Anything
	Horse
	Generate names
	Captain NobleStride, Steed of Valor, Mighty Equine Warrior,
	`openai-quickstart-python-master — python ◂ flask run — 80×24`
	`(EEE104-AI) shivanshshukla@Shivanshs-MacBook-Pro envs % cd /Users/shivanshshukla/Desktop/SJSU/Fall\ \'23\ \(last\)/EE104/Lab\ 6/Hello\ World\ to\ Open\ AI/openai-quickstart-python-master`
	`(EEE104-AI) shivanshshukla@Shivanshs-MacBook-Pro openai-quickstart-python-master % . venv/bin/activate`
	`(venv) (EEE104-AI) shivanshshukla@Shivanshs-MacBook-Pro openai-quickstart-python-master % flask run`
	` * Serving Flask app 'app' (lazy loading)`
	` * Environment: development`
	` * Debug mode: on`
	` * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)`
	` * Restarting with stat`
	` * Debugger is active!`
	` * Debugger PIN: 100-878-829`
	`127.0.0.1 - - [31/Oct/2023 19:46:56] "GET / HTTP/1.1" 200 -`
	`127.0.0.1 - - [31/Oct/2023 19:46:56] "GET /static/main.css HTTP/1.1" 304 -`
	`127.0.0.1 - - [31/Oct/2023 19:46:56] "GET /static/AI_server.png HTTP/1.1" 304 -`
	`127.0.0.1 - - [31/Oct/2023 19:46:56] "GET /static/AI_server.png HTTP/1.1" 304 -`
	`127.0.0.1 - - [31/Oct/2023 19:47:46] "POST / HTTP/1.1" 302 -`
	`127.0.0.1 - - [31/Oct/2023 19:47:46] "GET /?result=+Captain+NobleStride%2C+Steed+of+Valor%2C+Mighty+Equine+Warrior%2C HTTP/1.1" 200 -`
	`127.0.0.1 - - [31/Oct/2023 19:47:46] "GET /static/main.css HTTP/1.1" 304 -`
	`127.0.0.1 - - [31/Oct/2023 19:47:46] "GET /static/AI_server.png HTTP/1.1" 304 -`

Big Quiz Game	 Pygame Zero Game 1. What is the unit of electrical resistance? 9 Ampere (A) Ohm (Ω) Volt (V) Watt (W)
Big Quiz 'game over'	 Pygame Zero Game Game over! Your score is 8/8. Time's Up ~ ~ ~ ~

Big Quiz kernel	 `Clicked on answer 2`
	`You got it correct!`
	`Clicked on answer 1`
	`You got it correct!`
	`Clicked on answer 4`
	`You got it correct!`
	`Clicked on answer 3`
	`You got it correct!`
	`Clicked on answer 2`
	`You got it correct!`
	`Clicked on answer 4`
	`You got it correct!`
	`Clicked on answer 1`
	`You got it correct!`
	`Hint: correct answer is box #3`
	`Clicked on answer 3`
	`You got it correct!`
	`End of questions`

VIDEO RECORDINGS

Recording Title	URL	Notes
CNN Model	https://youtu.be/_3ITQG-hgtg	CNN Model Code Demonstration
Test Image	https://youtu.be/WXO71ByDOzI	CNN Test Image Demonstration
Big Quiz Game Development	https://youtu.be/ygVPTKlmxpo	Big Quiz explanation with Tweaks
EE104 - Lab 6: Hello World to OpenAI	https://youtu.be/LErjWz_bAQg	Demo for OpenAI website that suggests pet names
EE104 - Lab 6: Hello World to ChatGPT	https://youtu.be/_LiI_McmEPk	Demo for ChatGPT loop query

CONCLUSIONS

In conclusion, this lab was undertaken with the overarching goal of gaining a comprehensive understanding of Convolutional Neural Networks (CNNs) and their applications, particularly in image recognition, as well as exploring various machine learning and AI projects. The team successfully achieved the objectives laid out in the initial plan. We not only recognized the fundamental principles of CNNs and their architectural components but also effectively implemented and improved upon a baseline CNN model, achieving the desired test accuracy. Furthermore, we ventured into diverse projects, from creating a game with interactive elements to developing web applications that interacted with AI models, thus broadening our skills and experiences.

While this lab can be deemed successful, it has also offered valuable lessons. First and foremost, comprehensive documentation and organized version control through GitHub have proven indispensable for effective collaboration and knowledge sharing. Furthermore, the ability to leverage cloud-based resources and robust hardware, particularly for AI models, can significantly enhance the efficiency and performance of these projects. As we reflect on this lab's achievements, we look forward to optimizing our workflows by harnessing better resources and continuously exploring new techniques and technologies in future endeavors.

REFERENCES

- What Is Agile Reporting? (Definition and How To Complete) https://www.indeed.com/career-advice/career-development/agile-reporting
- Test Report https://strongqa.com/qa-portal/testing-docs-templates/test-report