WEB BASED DOG BREED CLASSIFIER: A COMPARISON OF CNN AND GPT-4O (CONTINUATION)

How does a multimodal Large Language Model compare to a Convolutional Neural Network when classifying dog pictures into breeds using user uploaded images?

Hannah Graham [1, 2,†]

1 Northwestern University

2 Code Repo:

Live Web Application: https://hannah-r-graham.github.io/DogBreedClassifier_whoDidItBetter/

Front end: https://github.com/hannah-r-graham/DogBreedClassifier_whoDidItBetter

Back end: https://github.com/hannah-r-graham/DogBreedBackend

† Email: Hannahrg24@gmail.com

Abstract

Ever look at a dog and wonder what breed it is? This research explores the development, comparison, and deployment of two distinct models for determining a dog's breed based on an image. The primary models to be used are a Convolutional Neural Network (CNN) with a 92.5% accuracy rate and the Large Language Model (LLM) GPT-4o, created and maintained by OpenAI. The objective is to enable users to upload an image of a dog and receive an accurate breed prediction, through a website running on Microsoft Azure.

The study builds on prior work with CNNs, which have been effective in image classification tasks since the 1980s (Kingler 2024). In contrast, Large Language Models like GPT-4o, which became popular in 2023, are designed to understand and generate human-like text. This research aims to integrate these two technologies to enhance breed prediction accuracy and user experience. The web application is deployed at: https://hannah-r-graham.github.io/DogBreedClassifier_whoDidItBetter/ , with both GPT-4o and CNN providing predictions. After deployment, the CNN maintains its 92.5% accuracy rate, as well as GPT-4o at 83.3% accuracy. However, given realistic sample images outside the original data set, GPT-4o performs better at recognizing only the dog breed in the image and ignoring people, while the CNN struggled to do so and occasionally misclassified the breed. More systematic testing is needed here with diverse dog images.

Table of Contents

Introduction and Problem Statement

How do we teach a computer to determine a dog's breed just by intaking a picture? How do we know which model is the most accurate method for classification? This scenario will enable users to upload an image of a dog and get a prediction back for the dog breed using multiple models. Classification will use a Convolutional Neural Network (CNN), a developed supervised classification model with a 92.5% accuracy rate, as well as the Large Language Model (LLM) GPT-4o, created and maintained by OpenAI.

CNNs were first developed in the 1980's for use with recognizing handwritten digits and later gained popularity for their success with the AlexNet model in 2012 (Kingler 2024). However, Large Language Models recently gained popularity in 2023, with the ability to intake text and images on March 14, 2023, OpenAI (2023). CNNs were made for classification and are a part of the Machine Learning category. GPT-4o is a generative model, attempting to understand user intent and predict the next phrase a person would want. These two technologies are very different in how they are built. This paper explores how well these two models can work together to enable users to understand a dog's breed through images.

This research is an expansion on a previously developed Convolutional Neural Network (CNN) and prompts using GPT-4o to determine dog breeds based solely on images. This research will build on the prior work in two constructive ways:

1. Develop a web application running on Microsoft Azure where users can upload a picture of their dog and both models give a prediction of the dog breed
2. Improving the GPT-4o prompt's current accuracy of 83.3% to meet or exceed the 92.5% accuracy of the CNN.

Literature Review

Convolutional Neural Networks have been in use for decades and show promising behavior when classifying dog breeds through images, with an accuracy up to 90% even in casual environments. Large language models are also now being employed to perform and assist in image classification. LLMs such as GPT-3 and GPT-4o are used to generate descriptions which are then fed into another model, or in some cases, generate not only a description but also a preliminary category for the image. Although CNNs have proved their worth, LLMs also

show potential due to the flexibility in how they are applied to image classification scenarios. How do these two very different models compare with image classification?

Classifying the dog breed in an image via CNN is a popular venture. Subin Thomas (2023) and Tuan Nguyen (2019) performed two separate experiments to obtain the same goal: take images of dogs and classify their breed using a CNN. Thomas was much more successful with a 90% accuracy rate on the validation data set, while Nguyen achieved only an 81% accuracy rate. The difference in accuracy rate could be due to differences in shape, Nguyen chose 224 x 224 while Thomas chose 350 x 350. Both used epochs of 5 and different batch sizes of 32 (Thomas) and 20 (Nguyen). They used different data sets. These variables could all contribute to the 10% difference in accuracy. Much of the CNN construction of this research leverages the work and code of Subin Thomas, much deserved acknowledgment of their work is recognized here.

While Large Language Models gained more popularity recently than CNNs, LLMs are already being leveraged to support image classification through gathering descriptions at scale, despite being a generative model and not a classification trained model. In a joint effort by University of Washington, Google DeepMind, and ML Collective, the team leveraged LLMs to create expansive descriptions of characteristics of image categories, then an open vocabulary model uses the description as prompts for the classification. These descriptions allowed the model to emphasize certain elements in the image to help determine a more accurate label, giving the model a more specific “direction” to pay attention to, so to speak, when determining an image class according to Pratt et al. (2023). An illustration explaining Pratt et al.’s work directly from their research can be found below in Figure 1. An illustration of Abdelhamed et al.’s interpretation of Pratt et al.’s work can be found in Figure 3.

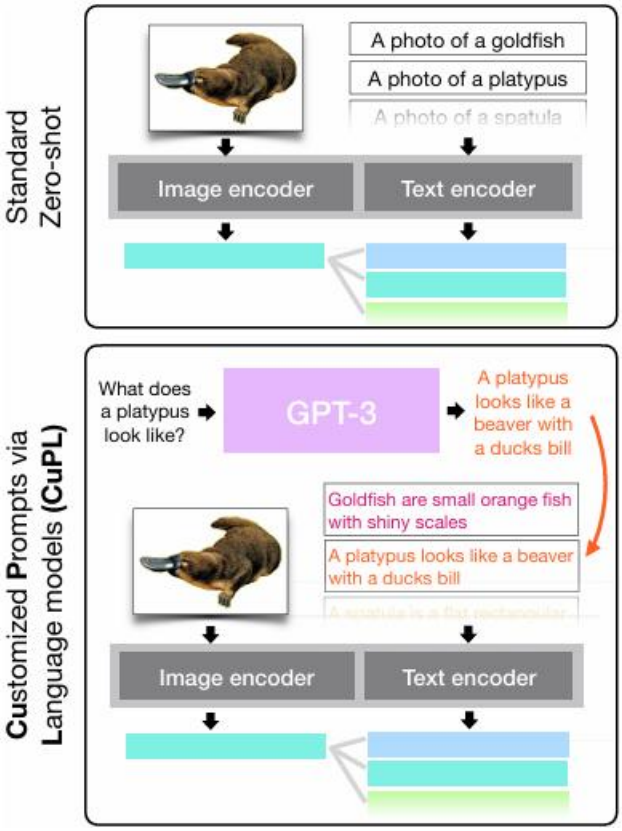

Figure 1: A comparison of standard zero shot learning and the work completed by Pratt et al. through Customized Prompts via Language Models (CuPL). Source: Pratt et al. 2023

Columbia University did similar work to Pratt et al. except with Vision-Language models, by using GPT-3 to create descriptions of certain categories and having the VLM check for those descriptors when classifying the image (Menon and Vondrick, 2022). Figure 2 displays how the team leveraged descriptions and weights for those descriptions then compared them with a common visual model CLIP.

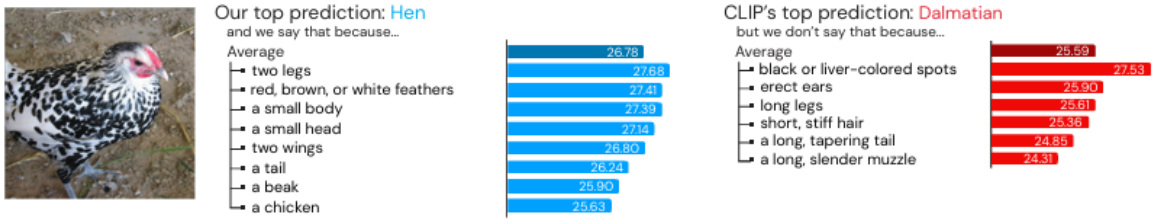

Figure 2: Visualization of Menon and Vondrick, 2022 work of using GPT3 generated descriptions to help with image classification. The figure compares Menon and Vondrick 2022's work with a visual model CLIP. Source: Menon and Vondrick, 2022.

Google Research has taken descriptions with LLMs a step further. They published a study of their findings using LLMs for zero-shot image classification in October 2024, only a month prior to this paper's creation. The team used a single set of prompts that can be used across various datasets to gather descriptions about an image and an initial suggestion of the class of the image. Then the image and the LLM description are then fed into a standard zero-shot linear image classifier. In other words, the LLM is used to create more data to help inform and support the zero-shot image classifier to make a more accurate prediction of the image Abdelhamed et al. (2024). A visualization of their methods as compared to CuPL Pratt et al. 2023 can be seen in Figure 3 below.

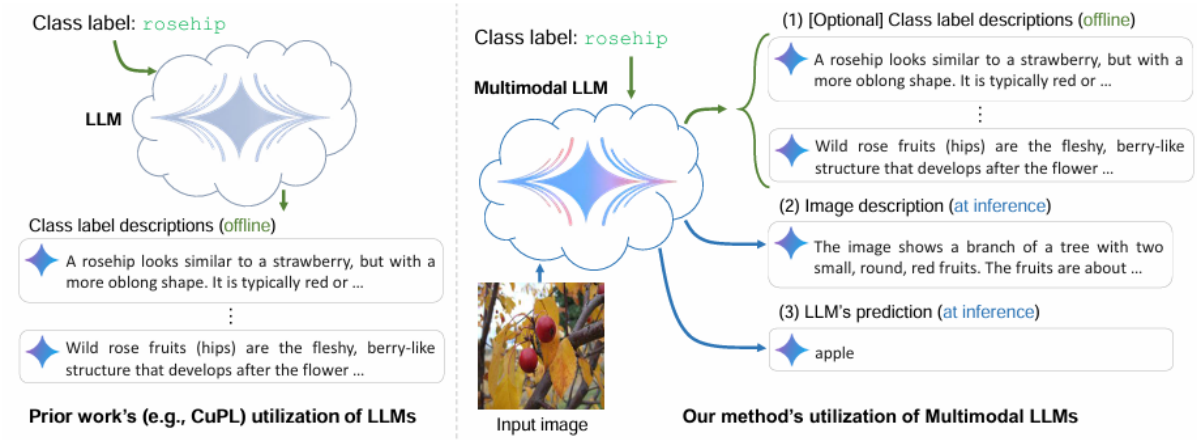

Figure 3: A visual representation comparing Pratt et al.'s work and Abdelhamed et al.'s work through the lens of Abdelhamed et al.. The team leveraged LLM generated descriptions in addition to an LLM determined prediction on the item to classify the image where Pratt et al. only used descriptions from the LLM. Source: Abdelhamed et al. 2024.

Data

Source data is from Kaggle, a well-known data science competition website. This data set was originally used in a classification competition in 2018. The data contains 120 breeds of dogs and 10,357 different images of dogs in jpeg format. Files include: one csv that has an

image ID and the dog breed label for that image. The second file contains JPEG images of all dogs, and the name of the image is the image ID correlating to the first file with breed labels.

Each breed has an average of 85.18 images. The Scottish Deerhound has the most images in the dataset at 126. The Briard dog breed has the least at 66.

count	120.00 breeds
mean	85.18 images per breed
std	13.29 images per breed
min	66.00 images per breed
25%	75.00 images per breed
50%	82.00 images per breed
75%	91.25 images per breed
max	126.00 images per breed

Table 1: Count of images per dog breed from the Kaggle dataset. These images will be used to help test and find the prompt that yields the highest accuracy from GPT-4o.

Min width	97 pixels
Max width	3,264 pixels
Min height	103 pixels
Max height	256 pixels

Table 2: Image sizes for the original dataset.

Methods

This research is an expansion on a previously developed Convolutional Neural Network (CNN) and prompts using GPT-4o to determine dog breeds based solely on images. The goal is to 1). Develop a web application running on Microsoft Azure where users can upload a picture of

their dog and both models give a prediction of the dog breed. Prior methods and new developments and processes are marked below with "(new)". 2). improve the GPT-4o prompts to meet or exceed the 92.5% accuracy of the CNN, improving the current 83.3%. Prior methods and new developments and processes are marked below with (new).

Overall process to be expanded in further detail:

1. Develop the CNN (completed already with 92.5% accuracy):
    a. Pre-process data for CNN to intake
    b. Train the CNN model, evaluate, then save the model.
    c. ensure the saved model is saved first with SaveModel command in TensorFlow. Then convert said file to a graph model file using TensorFlow converter so that it can be utilized by the web application. (new)
    d. Save CNN graph model file to Azure Blob storage for retrieval by the web application when deployed. (new)
2. Improve the GPT-4o prompts to obtain an accuracy of 92.5% or better on a sample 120 images, an increase from the current 83.3% accuracy. Sample size is limited due to the cost of running GPT-4o. (new)
3. Deploy both models to Microsoft Azure. (new)
4. Develop a front-end interface where users can upload images of their dog and get a predicted dog breed from both the CNN and GPT-4o. (new)
5. Store images uploaded from users in blob storage for 30 days. (new)

Develop the CNN (completed prior to this research):

Prepare the data for the CNN model:

Using Pandas and the CV2 package, the labels dataset and the images were accessed. For each image, the size of the image was stored then the min and max pixel size for each image was printed. Given the data, images were all resized to 350x350 pixels before being ingested by both the CNN and LLM. The images were split into training, test, and validation groups with an 80-10-10 split with a random state value of 42 using the SKLearn package.

Build the model:

The Convolutional Neural Network (CNN) was constructed using the Xception model pre-trained on ImageNet as the base model. The model architecture included a global average pooling layer, a dropout layer with a rate of 0.7, and a dense output layer with softmax activation for classification into the number of unique dog breeds.

Two callbacks were employed during training: EarlyStopping and ModelCheckpoint. EarlyStopping monitored the validation loss and stopped training if it did not improve for four consecutive epochs. ModelCheckpoint saved the model with the best validation loss.

Model training and evaluation:

The model was trained using the fit method with data generators for training, validation, and test sets. The training process included 5 epochs with a batch size of 32. The model's performance was evaluated on the test set with an accuracy of 92.5%.

LLM methods:

Choosing a Large Language Model:

GPT-4o is OpenAI's largest and one of their most recent models and therefore the chosen model for this research. Access to this model is through Azure OpenAI subscription, then through API Key and Endpoint accessed by the code through a .env file

Prepare the data:

For cost reasons, only 120 images were inputted into the LLM, the same 120 images as inputted into the CNN model after it was trained. These 120 images were resized to 350x350 pixels, like the CNN model. Using the base64 package, the JPEG images were to base64, 'ASCII'. UTF-8 was first attempted instead of ASCII, however the token count was 100x more for UTF-8. The Azure AI playground uses ASCII for images, so doing this conversion ahead of time reduced the token count and prevented the code from hitting the current rate and quota limits for Azure OpenAI API.

Web application deployment:

On Azure, a web application service is created and deployed to act as the back-end server for the application. A blob storage database is deployed through Azure where both the CNN model will be saved and retrievable by the web application and where images the users upload will also be saved for 7 days. The front end is hosted on GitHub Pages due to its low cost. The application uses React.js as its framework, and Node.js for the back end. Both the front end and back-end use GitHub actions for automated and rapid deployment to production.

Results

Web application:

The web application was deployed onto GitHub pages for the front end and Azure Web application for the back-end server side. The CNN model and images uploaded by users were stored in Azure blob storage where the application can then access them. The application was developed on React.js framework for its ease of use and flexibility for mobile and online applications.

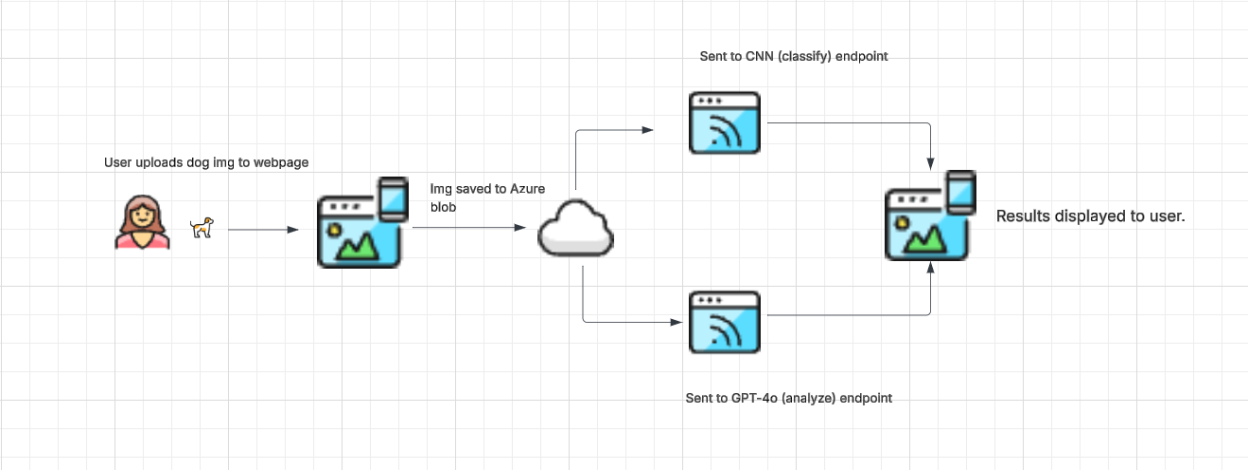

Figure 4: overall flow of the web application.

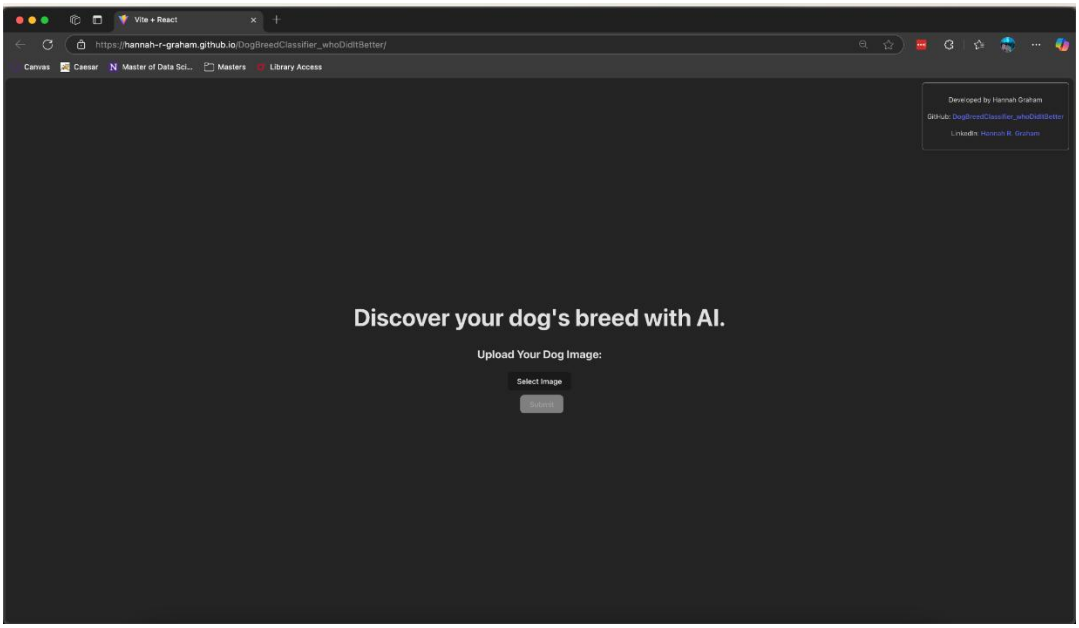

Figure 5: welcome screen of the web application. This is the first page users see.

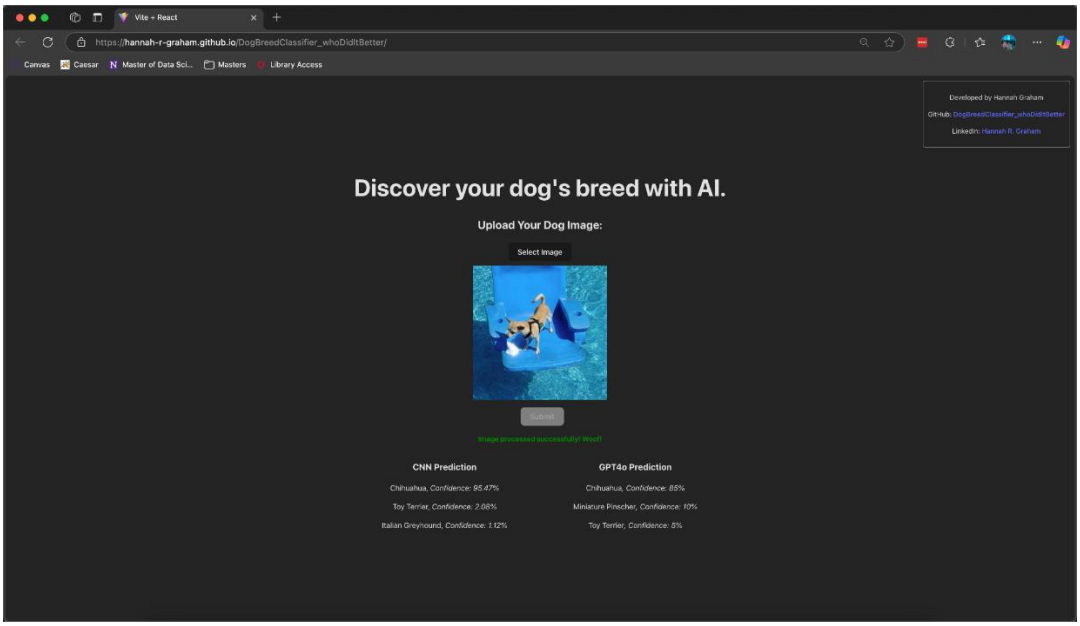

Figure 6: Results screen of the web application that shows the top 3 predicted breeds for both CNN and GPT-4o as well as their confidence intervals.

Prompt improvements:

Table 3: Results of prompt experiments. The same 120 images were used as Graham, 2024.

Prompt	***Model***	***Temp***	***Accuracy***
Prompt A (Graham, 2024)	*4o*	*0*	*83.3%*
Prompt B (asks for one breed + confidence)	*4o*	*0*	*83.3%*
Bonus Experiments on 4o-mini and temperature affects accuracy.			
Prompt C	*4o-mini*	*Not set*	*54.17%*
Prompt C	*4o-mini*	*0*	*57.5%*

Prompt C (asks for top 3 breeds + confidence)	*4o*	*0*	*83.3 %*

Language of the most accurate prompts (both with an 83.3% accuracy):

1. Prompt B:
    a. *"text": Determine which dog breed is in the input_image given the list of breeds below. Give the most likely breed, with a percent confidence. The output should look like: 1. breed, Confidence %. If you are unsure, put your best guess. Ignore any background and only focus on the dog. Breed list: {', '.join(breed_list)}*
2. Prompt C (used in production on web application):
    a. *"text": Determine which dog breed is in the input_image given the list of breeds below. Give the 3 most likely dog breeds in order of likelihood. Include a percentage likelihood for each breed. The output should look like: 1. breed, Confidence %, 2. breed, Confidence %, 3. breed, Confidence %. If you are unsure, put your best guess. Ignore any background and only focus on the dog. Breed list: {', '.join(breed_list)}"*

Important to note: Graham 2024 used Azure OpenAI API to connect to 4o while this paper used OpenAI API directly. There seems to be no difference between models or results, but it is still worth noting for future work.

Discussion

Web application development and deployment:

The most important takeaway from this stage is ensuring if using "SavedModel" function with Tensorflow.js to save the CNN locally, that you then convert it into TensorFlow.js format for a graph model that can then be used in the web browser environment. That graph model folder can then be used for local development of web applications. To use the CNN on the deployed application, the graph file needs to be saved to blob storage for the web application to be able to access it. Using the "SavedModel" saves the model as a ".layers" file and is not compatible with full-stack development.

Prompt Engineering:

Consistent with Graham 2024, CNN still has the highest accuracy at 92.5%, while 4o, despite changes in model (4o and 4o-mini), temperature, and prompt verbiage never surpassed the accuracy of 83.3% using a set of 120 sample images with 2-3 new images from personal, realistic use. The prompt was modified from Graham 2024 to ask for 3 top dog breeds and the confidence score, while Graham 2024 only asked for one dog breed and did not request a confidence score. When the prompt to ignore the background of the image and only focus on the dog, the accuracy score on the 120 sample images did not change, however, it did become more accurate on a random image selected from testing data (the chihuahua in the pool) and a personal image including a person. More details on this in "interesting cases" below. More testing is required, but overall CNN remains the most accurate with the 120 sample images.

A few interesting cases:

Backgrounds with interesting textures: The CNN did not struggle with the image of the dog in the pool in figure 7, or the dog in the grass in figure 8 which makes sense because these images were part of the train/test split for the CNN. But GPT-4o did struggle to identify the dogs. To help 4o pay attention to the dog, adding the phrase "*Ignore any background and only focus on the dog"* made GPT-4o recognize the chihuahua on the next try and the dog in the grass every test on these images since.

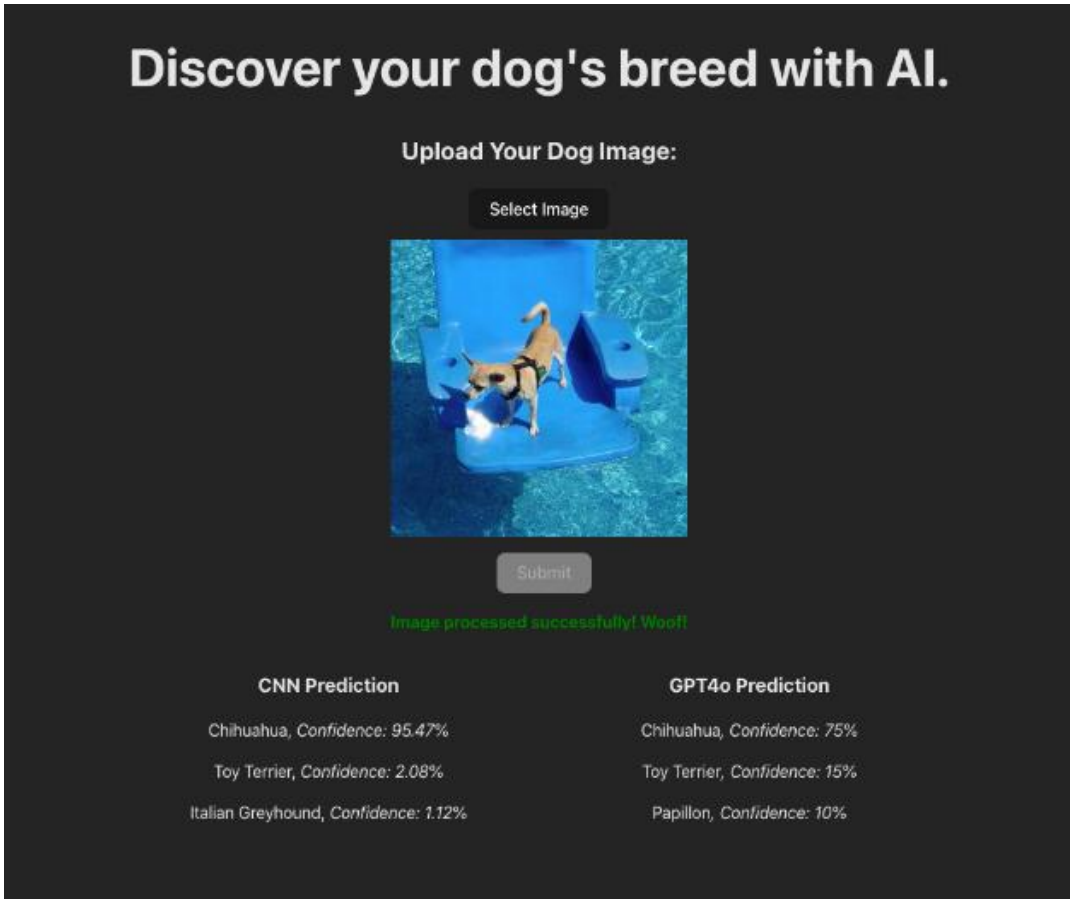

Figure 7: Interesting case of Chihuahua in a pool that lead to prompt improvements

Figure 8: Photo of dog in grass that also lead to prompt improvements along with Figure 7.

Containing people: The image below shows a dog on a person's lap. In a rare case so far in testing, the CNN's prediction did much worse than GPT-4o. 4o was able to discern the dog from the human and focus on the dog much better than the CNN. The CNN was trained on images generally only containing dogs and backgrounds (grass, pools, etc.) but rarely people. Hence its confusion and identifying the dog as an Eskimo dog. This leads to the theory that the CNN performs better when provided with quality, individual images of dogs, but when adding in diverse images that emulate more real-life scenarios with distracting backgrounds, 4o is more accurate. However, more wide scale tests are needed to prove this theory.

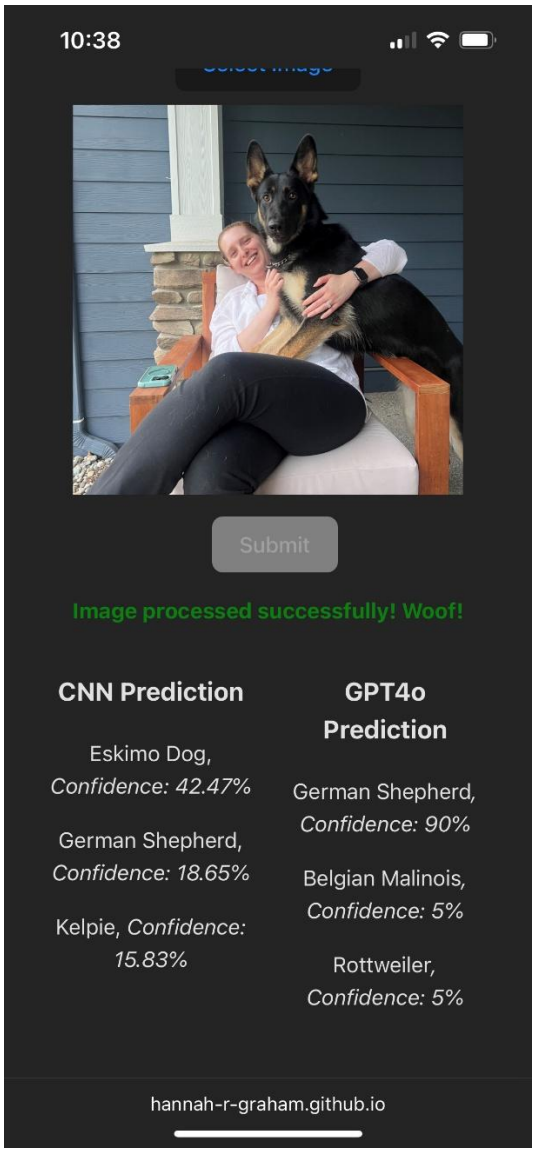

Figure 9: Personal photo uploaded to the tool.

Bonus findings:

When testing prompts, not including temperature setting versus including a definitive "0" for temperature, yielded 3% better accuracy for GPT-4o-mini than not including a temperature setting. This shows the importance of setting the temperature explicitly in the parameters when doing image analysis with an LLM. Temperature is a way to control how "precise" the LLM is versus more creative. The higher the temperature, the more creative the response (OpenAI, 2024). With image analysis, we want the most accurate prediction possible, so a temperature of 0.

Controlling for the same prompt and temperature setting but comparing models GPT-4o and GPT-4o-mini, 4o had a 29% more accurate result on the 120 sample images. While 4o-mini is cheaper and faster than 4o, in this case, accuracy was sacrificed. In addition, no noticeable differences in speed were detected for this size of data set.

Conclusions

When comparing models on image classification of dog breeds, a CNN is both cheaper and much more accurate than GPT-4o. CNN is easier to deploy and maintain, gives a standardized output that can easily be ingested by the API, and does not require additional API calls to sources outside the web application. A web application is a great way to demonstrate and compare these two models while also displaying the power of Data Scientists gaining full-stack software engineering skills. However, from a user perspective, many of the images the CNN is trained on are "ideal samples" and won't be reflected in the type of images the end user uploads. In some cases, though further study is needed, such as humans in the image with the dog, 4o did a much better job than the CNN at ignoring the "noise" in the image and outputting a useful result.

Regardless of model output and effectiveness, the ability for models to be implemented in web applications that users can then interact with in a common way is key for Data Scientists to help improve the model to product pipeline. The more Data Scientists understand how their models are being used and implemented, the more practical the models and software development pipeline becomes.

Directions for Future Work

Enhancing the web application involves several key improvements. First, improving accessibility will make the application more inclusive, allowing users with disabilities to interact with the platform effectively. Additionally, controlling access and usage is essential for managing operational costs and optimizing resource allocation.

In terms of model development, it would be ideal to enable analysis of mixed breeds and provide estimated percentages for each breed. This feature would make the application much more usable by people as mixed breeds are more common than purebreds.

Acknowledgements

Data Availability

Data for original images can be found here: Dog Breed Identification | Kaggle

Code Availability

Live Web Application: https://hannah-r-graham.github.io/DogBreedClassifier_whoDidItBetter/

Front end: https://github.com/hannah-r-graham/DogBreedClassifier_whoDidItBetter

Back end: https://github.com/hannah-r-graham/DogBreedBackend

References

Abdelhamed, Abdelrahman, Mahmoud Afif, Alec Go. 2024. "What do you see? Enhancing zero-shot image classification with multimodal large language models". *arxiv.org.* (Accessed November 3, 2024; available online at https://arxiv.org/pdf/2405.15668).

Graham, Hannah. 2024. "Dog Breed Classifier: A Comparison of CNN and GPT-4o". GitHub.com. (Accessed May and June 2025). Available at DogBreedClassifier_whoDidItBetter/DataFiles/DogBreedClassificationAComparison_hagraham.pdf at main · hannah-r-graham/DogBreedClassifier_whoDidItBetter

Kingler, Nico. 2024. "Convolutional Neural Networks (CNNs): A 2025 Deep Dive". *Visio.ai.* (Accessed November 1, 2024; available at https://viso.ai/deep-learning/convolutional-neural-networks/).

Menon, Sachit, Carl Vondrick. 2022. "Visual classification via description from large language models". *arxiv.org.* (Accessed November 1, 2024: available at https://arxiv.org/pdf/2210.07183).

Nguyen, Tuan. 2019. "Build Your First Computer Vision Project — Dog Breed Classification". *Medium.com.* (Accessed November 1, 2024; available at https://towardsdatascience.com/build-your-first-computer-vision-project-dog-breed-classification-a622d8fc691e).

OpenAI. 2024. "GPT-4". (Accessed November 1, 2024; available at *https://openai.com/index/gpt-4-research/).*

Pratt, Sarah, Ian Covert, Rosanne Liu, Ali Farhadi. 2023. "What does a platypus look like? Generating customized prompts for zero-shot image classification". *arxiv.org.* (Accessed November 1, 2024; available at https://arxiv.org/pdf/2209.03320).

Thomas, Subin. 2023. "Deploying a Dog Breed Classification ML Model: An End-to-End Guide". *Medium.com.* (Accessed November 1, 2024; available at https://medium.com/@subin60/deploying-a-dog-breed-classification-ml-model-an-end-to-end-guide-fc7c025e13a2).

Cukierski, Will. 2017. "Dog Breed Identification". *Medium.com.* (Accessed November 1, 2024; available at https://kaggle.com/competitions/dog-breed-identification).

Appendix A