COMS6998 Final Project: AI Pictionary

Ericka Wu & Jonathan Zhang

Goal/Objective:

In this project, we will aim to construct and build a neural network with the ability to translate between natural language descriptions and 2D drawings. Recent state of the art systems that perform text-to-image and image-to-text synthesis have advanced greatly in accuracy and human-likeness, prompting interesting questions about the origins, extent and nuances of AI “creativity” and human-like behavior.

We find the interplay between text and pictures to be both a particularly difficult and visually appealing demonstration of Neural Network capabilities. In the “text-to-image” direction, our network must not only be able to effectively classify objects under the guise of idiomatic and uncommon prompts such as “otter eating sushi” or “hamster that’s trying its best,” identifying the key features of a sketch and parse a representative description.

Summary:

Although guessing, Pictionary-like games have been created powered by different CNNs, they are only able to classify objects based on the classes they are trained on. As we wanted to try a set of fun prompts, we knew that a general classifier wouldn’t be able to do the trick. Therefore, we harnessed the power of a state-of-the-art network that would learn generalizable text and image connections: CLIP. Then, we fine-tuned CLIP using data from Google’s QuickDraw dataset, which is available on a public Github repository. Lastly, to demonstrate the abilities of this network, we hosted a webserver that accepts user input on a drawing canvas: anything drawn can be evaluated by our fine-tuned model.

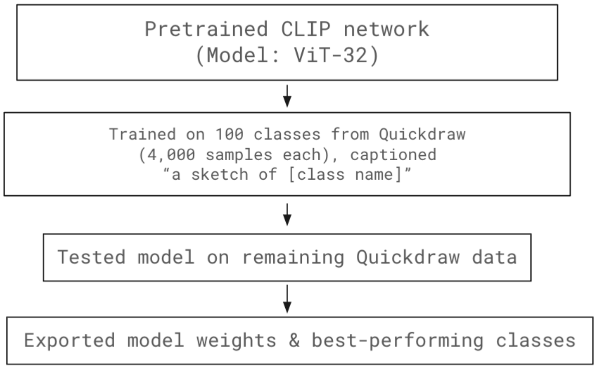
Fine-Tuning CLIP:

In order to properly scope the project, we limited interpretation features to black and white sketches. Not only do sketches contain a smaller capacity for information than full images, we believe that it’d be an interesting exploration of how a neural network can express and understand creativity. We used data from Google’s QuickDraw dataset, which is available on a public Github repository (linked in the References section). It contains 50 million drawings across 345 different categories, contributed by real-life players to Google’s game Quick, Draw! As this is quite a large dataset, we pulled 5,000 images each from 100 QuickDraw classes, totalling 500,000 28x28 images for our training and testing set.

CLIP can perform zero-shot prediction given an image and textual labels. CLIP is a pre-trained pair of networks, one image encoder and one text encoder, and it’s trained with a contrastive objective: instead of simply predicting text, for each mini-batch, it predicts which images are paired with which text captions after running them through their respective encoders. And, CLIP has been pretrained on 400 million text-image pairs publicly available on the internet. There are four available pretrained architectures: we chose the most advanced one to work with, which involved a language and a visual transformer. To predict a caption given a set of possible prompts and a set of images: for each image, the model runs the text and the image through their respective cutters and returns the most probable text match for that image.

We split the QuickDraw data 80-20% for testing and training. The baseline accuracy for CLIP turned out to be rather low, a figure of 10.68% across all 100 classes, as CLIP was pretrained on real images and not sketches. Therefore, we retrained the model with our subset of QuickDraw data using a batch size of 64 with the same contrastive objective for 50 epochs, achieving a final accuracy of 70.14%.

Below is a diagram of our model training and testing pipeline:

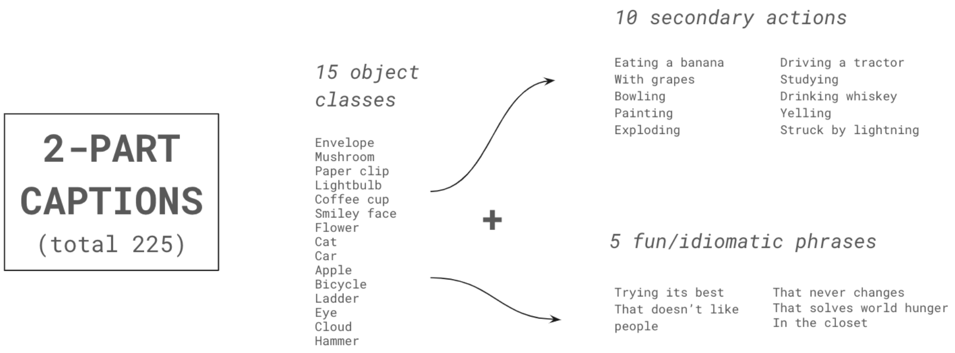

Pretrained CLIP network
(Model: ViT-32)
Trained on 100 classes from Quickdraw
(4,000 samples each), captioned
"a sketch of [class name]"
Tested model on remaining Quickdraw data
Exported model weights & best-performing classes

Prompt Generation:

One of our challenges was finding a way to assess the human-likeness of a model. Given our trained model, how do we look at how well it expresses creativity? To do this, we created our own set of fun and idiomatic prompts.

Each prompt was composed of two parts, the first being an object class and the second being either a secondary action or a vague phrase. After fine-tuning CLIP, we took the first 15 object classes that the model did best on, as well as composed 15 arbitrary secondary actions and fun phrases that could be added to those classes. We took all permutations of the classes and secondary phrases for a total of 225 fun prompts.

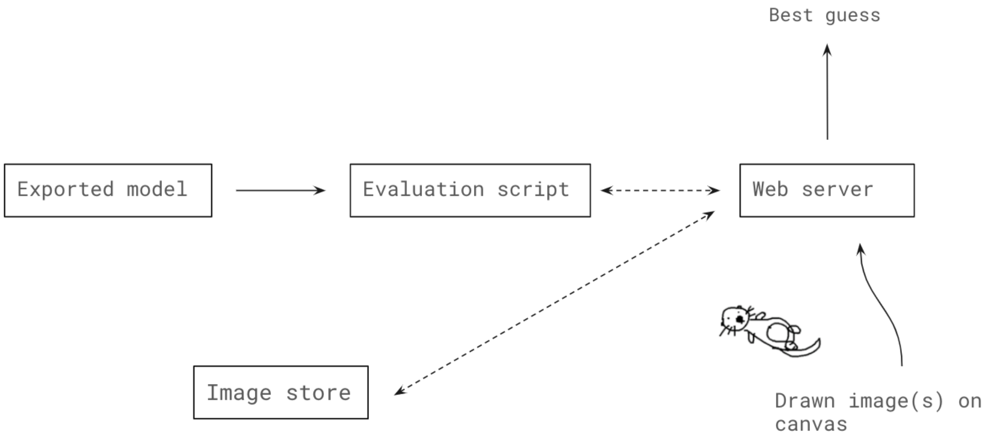
All of the words used for our 225 prompts are shown in the image below:



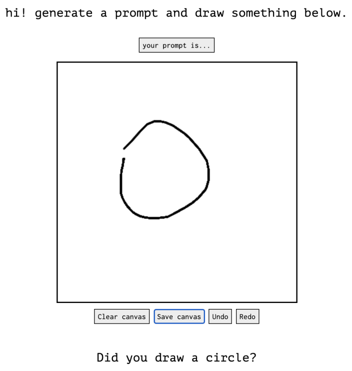
Demo Implementation Details:

We used the Google Cloud Platform GPU to host our webserver and train our neural networks, creating a simple web stack with a frontend canvas for user-submitted sketches.

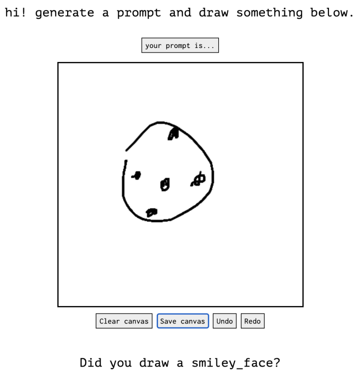
The basic architecture is shown in the following diagram:



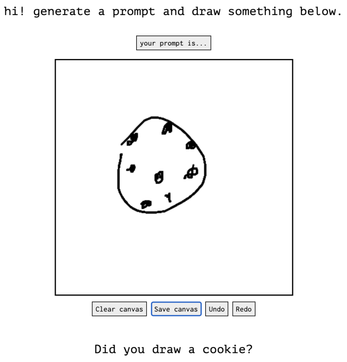
Visual Analysis:


hi! generate a prompt and draw something below.

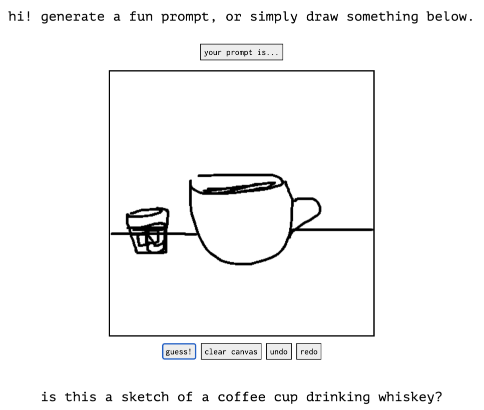
Did you draw a circle?


hi! generate a prompt and draw something below.

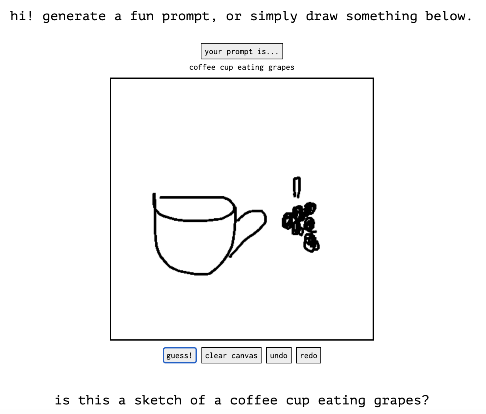
Did you draw a smiley_face?


hi! generate a prompt and draw something below.

Did you draw a cookie?


hi! generate a fun prompt, or simply draw something below.

is this a sketch of a coffee cup drinking whiskey?


hi! generate a fun prompt, or simply draw something below.

coffee cup eating grapes

is this a sketch of a coffee cup eating grapes?

Conclusion & Further Work:

References:

https://github.com/openai/CLIP: can predict the most relevant text snippet given an image
https://github.com/googlecreativelab/quickdraw-dataset: Google QuickDraw dataset