HungryAI - Final Report

Alder French
University of Oregon
afrench7@uoregon.edu

Abstract.
After realizing there was an apparent need for efficient image generation for training purposes, the HungryAI experiment began with the purpose of training a GAN on images of individual foods, and seeing if that GAN could then produce images of "combinations" of those foods that are useful to a classifier.

1. Introduction

My first final project was supposed to be an image classifier that took images of plants as input, and output a label that indicated whether that plant was healthy, or needed some attention. E.g. "Plant is healthy" or "Plant needs more water and sun". HungryAI started after looking for images of plants, with different apparent levels of health, and realizing there was not nearly enough to scrape off Bing images to realistically do a good job training a classification model.

I found this frustrating as I felt the plant health classifier was a useful application of ML technology. After thinking about where I could find data about plants' health, I realized that there is a lot of data in books in the form of text. This got me thinking. Humans are able to imagine things we've never actually seen before, often with the aid of text such as from books, so why can't AI?

To properly experiment with this concept, I figured I should start with a subject that has many easily recognizable components, as well as easily recognizable combinations of those components. By doing so, giving the GAN the best chance possible to generate images that would be useful for the classifier.

2. Experimentation

A. Find models for GAN and Classifier

After doing some research, I settled on using a GAN called SSA-GAN from the paper by W. Lao et al [1]. I chose this GAN model because it has "Semantic Spatial Aware" blocks that work with a text encoder to better understand how sentence structure provides context for an image. This is helpful when trying to generate images based on full sentences of text, rather than simple labels like "chicken and waffles". For my classifier, I chose the Resnet50 model as it is easy to work with, efficient,

and yet powerful enough to yield accurate classifications.

B. Data Collection - Images and Text

Next, I had to get a bunch of images of individual foods for training and combo foods for testing. To do this, I used an old image scraper python script of mine to scrape bing images. For the text, I used ChatGPT to generate 18 word or less descriptions of the individual and combo foods. There were 25 captions per class of 100 photos for the individual foods image GAN training, and 10 captions per class for the food image generation. Here's an example of a couple of the descriptions:
"Golden fried chicken perched atop fluffy waffles, drizzled with syrup."
"Small and shiny Black Beans with a smooth texture."

C. Training

To train the SSA-GAN, I realized I would also need to train a text encoder DAMSM for the individual food images and their respective captions. So I downloaded a repo by Sidward14 on github [2] and modified its code to work with my image dataset and the other github repo I downloaded for the SSA-GAN. Then I was able to use this first repo to train a text encoder as previously described. Note that I trained the text encoder for 100 epochs, with a batch size of 24.

Next, I was all set to train the GAN, so I modified the SSA-GAN repo, this is when majority of the coding work I did took place. There was a lot to change due to PyTorch updates and differences between my dataset and the one used for SSA-GAN, even though I made my dataset with a folder structure nearly identical to that of the SSA-GAN repo's. Once it was finally working, I trained the SSA-GAN with the individual food images, with 1 randomly selected caption out of the 25 for each class for each image as a label. There were 21 individual food classes, each something like "Chicken" or "Waffles". Note that I trained the SSA-GAN for 300 epochs with a batch size of 6, which took ~30 hours on my Nvidia RTX 3060ti.

D. Generation

Once I was done training the SSA-GAN, I checked that it could generate images of the individual foods that looked ok. Then, I used 10 1-sentence descriptions of the combo foods (generated by ChatGPT) as input to the SSA-GAN to make it generate 100

images for each combo class (10 images per caption).

Figure 1. “Chicken and Waffles” by GAN

Figure 2. “Tomato Soup and Grilled Cheese” by GAN

Figure 3 (below). “Steak and Potatoes” by GAN

As you can see, the generated food combo images do indeed look like food. They are obviously weird, but to me, they somewhat resemble the real combo food class images for which they are supposed to help train the classifier.

3. Results

I trained and tested the Resnet50 model for 100 epochs using only the images of the combo food classes and their class names, e.g. “chicken and waffles”, as labels. Then I tested it on real images of the combo foods, and was pleasantly surprised to get an accuracy of 26%. Considering there are 10 classes, this indicates that the classifier was indeed able to learn about the combo classes from the generated images.

However, I believe there are more tests I need to carry out to really see if the generated images are useful and how useful they are. One of these tests is training the Resnet50 with the generated images AND the images of the real individual foods. If the Resnet50 could then discern between the individual food classes and the combo food classes, it would have truly learned from the SSA-GAN’s generated combo food images,

and is not just recognizing the food images based on their individual food components.

Another relevant test would be to try and collect more descriptions for each combo food, say 10 instead of 25, and use these to generate more images of the combo foods to train the classifier with. This would be very practical from a computation standpoint, as it only took only 49 seconds to generate 1000 images from the 100 descriptions in my first test, so it would be easy to generate 10x the images without taking up too much time. The descriptions collection for this task would be more difficult, and using just ChatGPT to generate these descriptions could be limiting, as ChatGPT often generates descriptions that are very similar to one another. If the classifier's accuracy improved as the GAN uses more descriptions to generate more images, then that would perhaps show that ML models can learn from text and use what they've learned to 'imagine' things they've never seen before, just like humans do.

In conclusion, GANs can certainly be used to generate images based on descriptions of images they have never seen before. This demonstrates their potential in zero - shot learning problems. A big takeaway from this experiment is we should not be afraid to get creative with how we use ML models. They have the potential to solve problems outside of their regular use cases, and by pushing them to their limits we may discover that they have capabilities far beyond what we currently think is possible.

4. References

1. Wentong Liao et al. *Text to Image Generation with Semantic-Spatial Aware GAN,* 2021.

2. Sidward14 (2022). *https://github.com/sidward14/Style-AttnGAN*

3. ChatGPT by OpenAI. Used for code generation, these code blocks will have a comment indicating so, and general PyTorch questions.

4. Handy Article on how to compute proper layer sizes by Jake Krajewski : https://towardsdatascience.com/pytorch-layer-dimensions-what-sizes-should-they-be-and-why-4265a41e01fd

5. University of Oregon CS 472 - "Machine Learning" course taught by Humphrey Shi and Steven Walton: "Provided some of the code used in my HungryAI.ipynb, and properly introduced me to the world of ML." *NOTE:* I couldn't have made this project without their help, and I learned a lot from the class. Thanks a bunch! - Alder