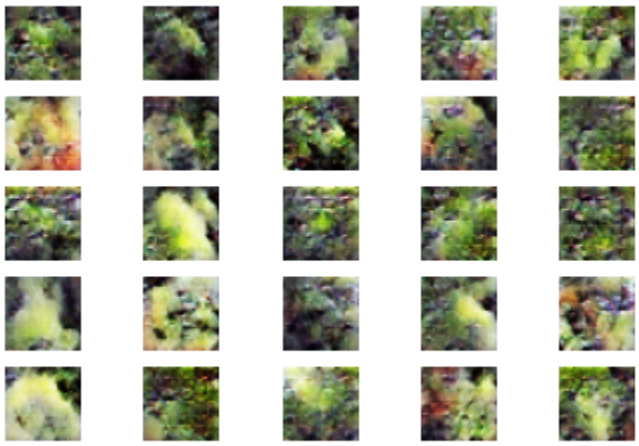
art + machine learning
cmu 2022 spring
project 2

new vegetables



group 28:

vivian young
--
Student-defined Major in Creative Technologies and Applied Computing
College of Fine Arts

description

"New vegetables" is a series of images generated by a dataset of ten classified vegetables and a Generative Adversarial Network (or GAN) implemented using TensorFlow in Python.

concept

The concept of this project was straightforward: after studying various ML algorithms and applications in class, I wanted to learn more about GANs and more specifically, how to implement them. After browsing various image datasets in Kaggle, I really liked the Vegetable Image Dataset and decided to train a GAN to generate images of pseudo-vegetables using this data. Food and color are prevalent themes within my artwork so this application seemed very cohesive with my prior body of work.

technique

Although I have a good amount of Python programming experience, TensorFlow and other Python ML libraries were very new to me. In addition, I still did not fully grasp the whole concepts of GANs so much research was needed in that area.

Ultimately, I found the following sources incredibly helpful in terms of walking me through their code and explaining the concepts behind each step:

https://livecodestream.dev/post/generating-images-with-deep-learning/

https://towardsdatascience.com/generative-adversarial-network-gan-for-dummies-a-step-by-step-tutorial-fdefff170391

https://machinelearningmastery.com/how-to-develop-a-generative-adversarial-network-for-a-cifar-10-small-object-photographs-from-scratch/

(My code is heavily borrowed from the last resource but is tweaked to be able to supply the dataset I chose which is hosted locally as well as some other small changes).

process

First I downloaded the aforementioned Vegetable Image dataset from Kaggle. I removed 5 classifications of vegetables (bottle gourd, brinjal, potato, papaya, and capsicum) because I either did not like those vegetables or they did not seem to fit with the rest.

For the code, I initially followed the first resource very closely. The initial segments of introducing the concepts and algorithm of GANs was very useful in understanding what I would be attempting to accomplish. However, I soon ran into many issues when I wanted to bring in my Kaggle dataset (stored in my local Desktop directory) as opposed to loading one of the built-in datasets in TensorFlow Keras.

example code:

```
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.fashion_mnist.load_data() # this is so easy
```

multiple attempts at trying to wrangle my folder of .png files into a similarly-formatted numpy array:

```
# process dataset files
data_dir = './data/'

# load train images
train_images = []
for filepath in os.listdir('./data/train'):
    train_images.append(cv2.imread('./data/train/{0}'.format(filepath),0))

train_images = np.array(train_images)
print(train_images)
```

```
# process dataset files
train_data_dir = './data/train' # my local vegetable data

BATCH_SIZE = 32
IMG_HEIGHT = 128
IMG_WIDTH = 128

train_datagen = ImageDataGenerator(rescale=1./255)
train_datagen_flow = train_datagen.flow_from_directory(train_data_dir,
                                                       target_size=(IMG_HEIGHT,IMG_WIDTH),
                                                       batch_size=BATCH_SIZE,
                                                       seed=42,
                                                       class_mode='categorical')

# load train images
train_datagen_flow.reset()
x_train=np.concatenate([train_datagen_flow.next()[0] for i in range(train_datagen_flow.__len__())])
print(x_train.shape)
```

```
train_images = tf.keras.preprocessing.image_dataset_from_directory(
    './data/train/', image_size=(224, 224), batch_size=64
)
```

This part gave me a lot of trouble but ultimately, I found my third resource that made a bit more sense to me so I changed my code to more closely match theirs.

```
def loadImages(pathname):
    numImages = len([name for name in os.listdir(pathname) if os.path.isfile(os.path.join(pathname, name))])
    print("Loading " + str(numImages) + " images from: " + pathname)

    result = np.zeros((numImages, 128, 128, 3))
    i = 0
    for file in os.listdir(pathname):
        filename = os.fsdecode(file)
        path = os.path.join(pathname, filename)

        result[i] = np.array(Image.open(path).resize((128,128), Image.LANCZOS),dtype=np.uint8)
        i+=1

    result = result.astype('float32')
    result = (result - 127.5) / 127.5 # normalize
    return result
```

The rest of the process involved a lot of copy-pasting of the code shown in the third resource, making small changes (such as changing the number of epochs, dimensions of the pyplot output, fixing lines of code due to updated python modules), and moving the code over to Google Colab.

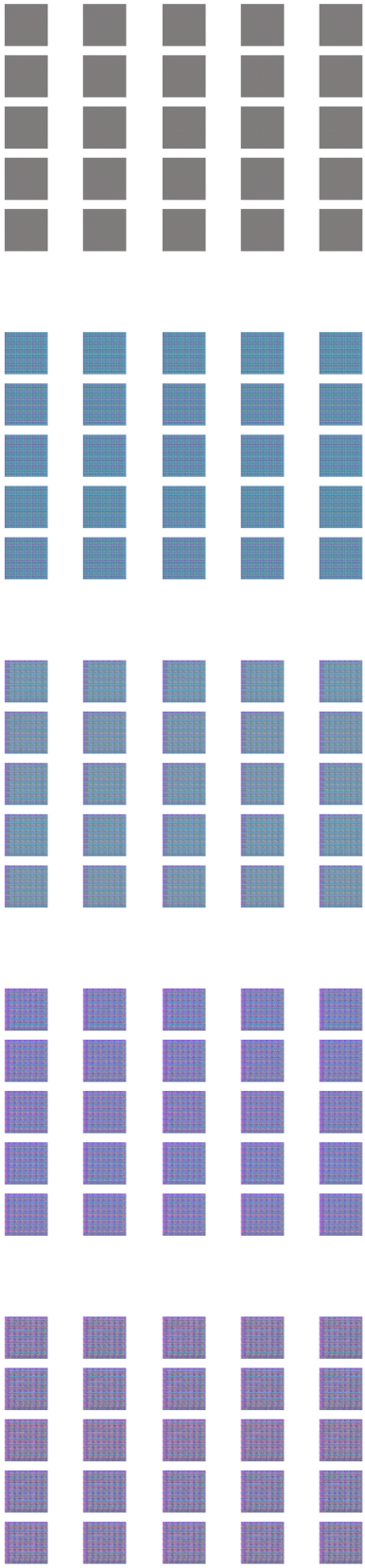
In order to upload my local dataset folder, however, I found that I needed to zip the dataset, upload the zip file to my Colab file, and run this code snippet so that my zipped dataset would be unzipped.

```
[3]  file_name = '/content/test.zip'

     with ZipFile(file_name, 'r') as zip:
       zip.extractall()
```

From there, I began training my GAN and saved the output images.

As an aside, on my first runthrough of training my model, I was experiencing very slow runtimes and could barely get through 10 epochs in 2 hours. I realized I had accidentally been analyzing the performance of the model and outputting an image at *every* epoch so my resulting images were very disappointing and the speed was super, super slow. Eventually, I was able to optimize the runtime by changing the dimensions of the images and calling summarize_performance(...) at every 5 epochs instead of every 1 epoch.



reflection
I am really happy at how much I learned throughout this entire process. I attribute almost all of my successes to the great resource I found as the code is almost entirely derived from the third source.

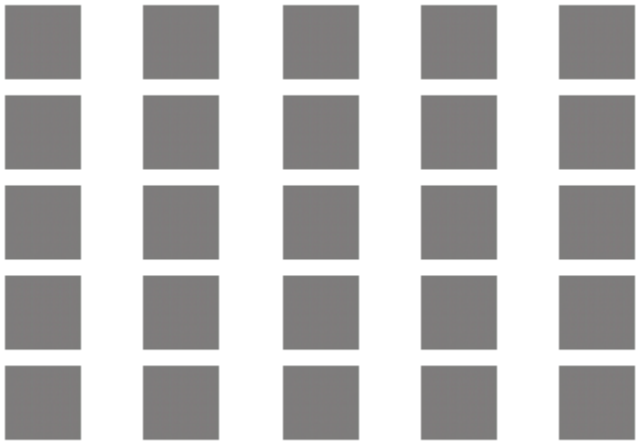
Ultimately, I selected the eventual image (the generated image made by the model after training 120 times) to represent this project because it was the most visually appealing to me and the closest representation of the concept I had in mind. I'm not sure it was as successful as I had hoped it would be and upon some later reflections, I believe this is because of the changes I made to the original dataset. Previously, the Vegetable Image Dataset provided train and test datasets which were both categorized into 15 subfolders containing the 15 different vegetables. I removed 5 of these subfolders (Capsicum/, Potato/, Papaya/, Brinjal/, and Bottle_Gourd/) because I didn't feel as though those vegetables made a lot of sense with the rest of the vegetables. What I did instead was take all of the ten types of vegetables in the training dataset and their 100 corresponding images each and put them all into my initial dataset. So now the dataset was no longer classified into categories and consisted of a mix of the ten types of vegetables, resulting in 10,000 images in total.

Perhaps I could improve upon my GAN by incorporating classification so that I'm generating new radishes, tomatoes, pumpkins, etc. individually instead of generating an entirely new vegetable that is an indiscernible mix of ten types of veggies. Let's consider my resulting image to be a new salad of sorts ... (haha).

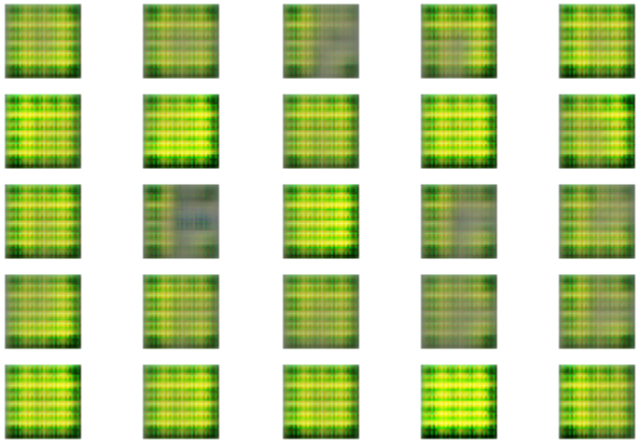
Moving forward, I hope to gain an even better understanding of GANs and utilize this technology in a variety of other media, such as sound or 3D modelling for example.

result

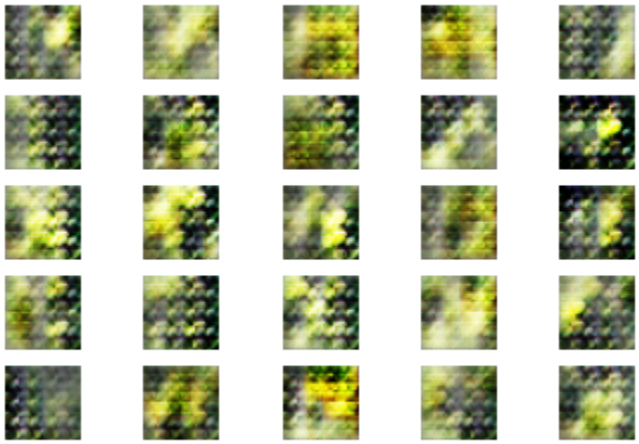
generated image after 0 epochs (untrained model):



generated image after 5 epochs:

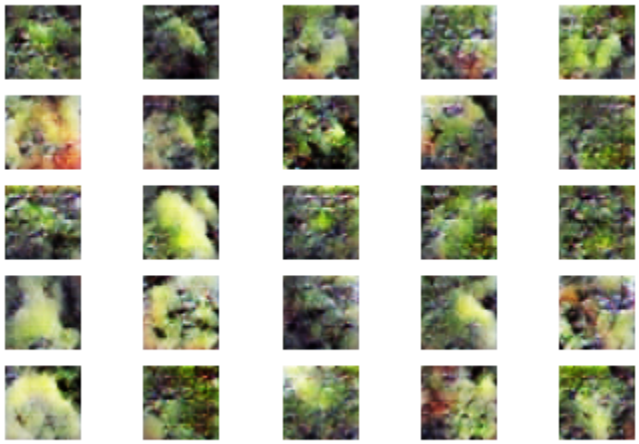


generated image after 50 epochs:



generated image after 90 epochs:

generated image after 120 epochs (selected image):



generated image after 125 epochs:

code

https://colab.research.google.com/gist/vivianyoung/952a8269c020101bba55f236909b24f4/new-vegetables.ipynb

reference

https://livecodestream.dev/post/generating-images-with-deep-learning/

https://towardsdatascience.com/generative-adversarial-network-gan-for-dummies-a-step-by-step-tutorial-fdefff170391

https://machinelearningmastery.com/how-to-develop-a-generative-adversarial-network-for-a-cifar-10-small-object-photographs-from-scratch/