https://drive.google.com/file/d/1s81JoId_3eEMQBQKmrk_2WpWCLGPqa3Z/view?usp=drivesdk - Demo
https://drive.google.com/file/d/13Js7fSw6WXKt2wXjEZ_MUSe7NM2K2kAp/view?usp=sharing - Presentation

This repository contains our course project `Generative Pretraining from pixels` of `GNR638` course `Machine Learning for Remote Sensing-11` done under prof. `Biplab Banerjee` in our third semester at IIT Bombay.

Teammates

- Hastyn Rajen Doshi - 200070025
- Neeraj Jadhav - 200050086
- Sarthak Mehrotra - 200010068
- Shrey Modi - 200020135
- Utkarsh Ranjan - 200050147

How To Run

The following is the directory structure of the project :-

```
(base) utkarsh@utkarsh-Spectre-360:~/Documents/iitb/gnr638/project/our_project$ tree
.
├── clusters
├── download.sh
├── execute.sh
├── Generative_Pretraining_from_Pixels_V2.pdf
├── models
├── notebook
│   ├── cifar10
│   │   ├── Image_gpt.ipynb
│   │   └── image_gpt_train.ipynb
│   └── mnist
│       └── igpt.ipynb
├── README.md
├── report.md
├── requirements.txt
├── results
│   ├── mnist.png
│   ├── result1.png
│   ├── result2.png
│   ├── result3.png
│   ├── result4.png
│   ├── result5.png
│   ├── test1.png.png
│   ├── test2.png
│   ├── test3_half.png
│   ├── test3.png
│   ├── test4_half.png
│   ├── test4.png
│   ├── test5_half.png
│   └── test5.png
├── sg.jpeg
├── src
│   ├── model.py
│   ├── __pycache__
│   │   ├── model.cpython-39.pyc
│   │   └── utils.cpython-39.pyc
│   ├── run.py
│   ├── utils.py
│   └── visualize.py
└── train
    └── k_means.py

9 directories, 31 files
```

- Main codes are in jupyter-notebooks:-
    1. **igpt.ipynb**: The notebook used to train the mnist dataset and obtain the results

```
Execute each cell one-by-one on google-colab/kaggle
```

2. **image_gpt_train.ipynb:** The notebook used to train the cifar10 dataset

```
Step 1: Download [kmeans_centers.npy]
(https://drive.google.com/file/d/1_F655q1DG0eKNSS7VfUoYQviz47E7Z7Y/view?
usp=sharing)
Step 2: Open the notebook and make a directory "clusters" at the remote
space of colab
Step 3: Upload kmeans_centers.npy in the directory "clusters" and
execute each cell
```

3. **Image_gpt.ipynb:** This notebook was used to get the ouputs from the trained model.

```
Execute each cell one-by-one on google-colab/kaggle
```

- Local execution:-

To run files locally follow these step:-

```
Step 1: Download [kmeans_centers.npy]
(https://drive.google.com/file/d/1_F655q1DG0eKNSS7VfUoYQviz47E7Z7Y/view?
usp=sharing) and put it in the clusters dir
Step 2: Download [models]
(https://drive.google.com/file/d/10ADYsVXjjkn_9YmpLwREvqlbcgW48vxQ/view?
usp=sharing) and put it in the models directory (0 level dir)
Step 3: pip install -r requirements.txt
Step 4: bash download.sh <image_link_to_be_executed>
Step 5: bash execute.sh <image_link_to_be_executed>
```

- Result directory contains the images generated.

Work Flow

The project can be divided into 4 parts broadly :-

1. **Preprocessing the image** : - First, we pre-processed raw images by resizing to a low resolution (28x28) and reshaping into a 1D sequence. We used opencv to convert crop images and reshaping it into the dimensions required to be fed into the model.
2. **Pre-training** :- This was done using two different dataset, one was done using **mnist** dataset and another using **cifar10** dataset. We used k-means clustering to find the color clusters to constitute our pixel vocabulary.
3. **Architecture** :- The image-gpt has similar architecture to the gpt-2 model where in we were learning embeddings from the pixel value of the image. We had 512 pixel values in our vocabulary and model was trained to learn embeddings for these pixel. We used a scaled down model of image-gpt with lessers num_layes, num_heads and num_embeddings due to training constraints.
4. **Fine-tuning** :- We average pooled across the sequence dimension to extract a d-dimensional vector of features. Further class logits were learnt to minimize a cross entropy loss.