Multi-domain inspired text-to-Artwork using CLIP-GAN

Course Project Report

Course: CS 726: Advanced Machine Learning

Students:

Sr. no.	Roll No.	Name	Email
1	190070006	Akshay Iyer	190070006@iitb.ac.in
2	19D070008	Amrit Rao	19D070008@iitb.ac.in
3	19D070035	Mihir Ghumbre	19D070035@iitb.ac.in
3	19D070056	Shubham Ojha	19D070056@iitb.ac.in

Code Link:

https://colab.research.google.com/drive/1niYvFIoJzLVZGrKGqwzA-bZgL-VSCuB1?usp=sharing

Problem:

In today's day and age, machine learning is getting increasingly important for daily tasks, and it is often used to make decisions that humans used to make, and perform human-like tasks. It is very important to understand the way a machine thinks and forms biases in the way it learns, so that we may be able to better understand them, and change their learning on observing any undesirable patterns.
The problem was first of all, to make a model that gave us the best description of an image described by a certain text. We do this using the CLIP-glass model (explained in the approach section), as decribed in [1]. We used scenery datasets, the labels of which were available, to train our CLIP-GAN model. Our main objective was to check the bias developed in this CLIP-GAN model.
We decided to use a CNN based on transfer learning, using weights from the ResNET50 model, in order to check the bias of these trained CLIP-GAN models. We did this by giving a subjective test query like "Beautiful landscape", to the algorithm, and got images from the CLIP-GAN model. We then classified the images into rivers, mountains, buildings, beach, etc. to check which scenery the algorithm found most beautiful.

Literature review:

0.1 Clip-Algorithm

CLIP (Contrastive Language–Image Pre-training) is a neural network model. It is trained on 400,000,000 (image, text) pairs. An (image, text) pair might be a picture and its caption. So this means that there are 400,000,000 pictures and their captions that are matched up, and this is the data that is used in training the CLIP model [3]. CLIP builds on a large body of work on zero-shot transfer, natural language supervision, and multimodal learning. The idea of zero-data learning dates back over a decade but until recently was mostly studied in computer vision as a way of generalizing to unseen object categories. A critical insight was to leverage natural language as a flexible prediction space to enable generalization and transfer. This is done by training a model to make predictions in a word vector embedding space, which could also predict unseen classes.

Thus CLIP is a pre-trained model, that predicts the similarity of an image to a given set of text, no matter if that type of image has been seen before or not.
We have taken inspiration from [1] where the CLIP-Glass algorithm is used, in order to generate the most adequate image described by an input text string. This algorithm checks the similarity scores of the generated image to the caption, for various different noise inputs to the GAN, and chooses an appropriate noise input, in order to generate the most suitable image.

0.2 DC-GAN

DCGAN uses convolutional and convolutional-transpose layers in the generator and discriminator, respectively. It was proposed by Radford et. al. [4] in the paper Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks. Here the discriminator consists of strided convolution layers, batch normalization layers, and LeakyRelu as activation function. It takes in a 3x64x64 input image. The generator consists of convolutional-transpose layers, batch normalization layers, and ReLU activations. The output will also be a 3x64x64 RGB image.

0.3 Style-GAN

The Style Generative Adversarial Network, or StyleGAN for short, is an extension to the GAN architecture that proposes large changes to the generator model, including the use of a mapping network to map points in latent space to an intermediate latent space, the use of the intermediate latent space to control style at each point in the generator model, and the introduction to noise as a source of variation at each point in the generator model.
The resulting model is capable not only of generating impressively photorealistic high-quality photos of faces, but also offers control over the style of the generated image at different levels of detail through varying the style vectors and noise.
Church Style-GAN is an a Style GAN whose training dataset consists only of images of churches. We thought it would be very interesting to see how a limited domain model generates images of unseen objects when used with the CLIP algorithm.

0.4 Big-GAN [2]

This architecture builds on top of SAGAN(Self-Attention GAN), which consists of self-attention mechanisms to refer from the details of the complete image. The self-attention block is denoted as the "non-local block" in the paper. This GAN shows that simply increasing the batch size and scaling the model can dramatically impact image quality. The latent vector Z and the class embedding y are passed through the generator with class-conditional BatchNorm in multiple scales and help enforce class consistency in the generated image.

Approach:

There were 2 aspects to this project. In one, we built our own GAN architecture, using a DC-GAN model, and trained it only on our scenery image dataset. We then used the CLIP-glass algorithm to display the best picture according to the text description. In the second approach, we did this exact same thing, except that we used a pre-trained GAN model, and just loaded its weights into our program.
We did this by using the CLIP algorithm developed by openAI, in conjunction with a GAN constructed by us. We used the CLIP algorithm to assign scores to images based on a given text. The way the CLIP algorithm works is by assigning the a score to the similarity, and dissimilarity of an image with an idea conveyed by a text string.
We constructed a GAN, that takes as input an input vector z, which is typically random gaussian noise. We used the CLIP model to tell us the similarity of the generated image with a predefined text description. Using this similarity score for a certain input z vector, we changed the input z vector in order to get an image which had highest similarity with the assigned text, as decided by the CLIP algorithm. This is called the CLIP-glass algorithm.

The way we decided which input z vector will give the best image, was by formulating an optimization problem and solving it using the Pymoo library.

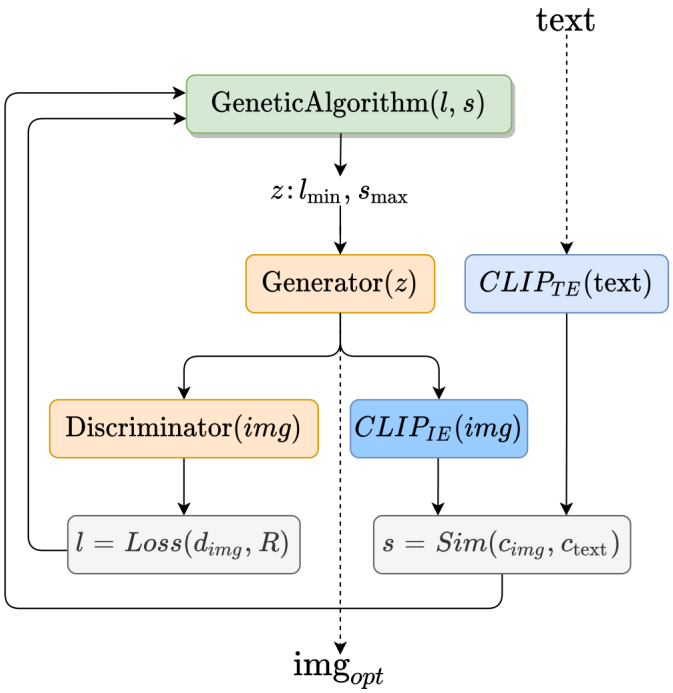

Figure 1: Working of Clip-GAN algorithm

The Pymoo library mainly works to iteratively optimize the 2 following problems-

$$\max_z sim(CLIP_{IE}(G(z)), CLIP_{TE}(T)) \tag{1}$$

$$\min_z Loss(D(G(z)), R) \tag{2}$$

Thus we find the optimal input z through the above optimization problem, to get the best output image generated by our model. Our GAN parameters and weights remain unchanged throughout this optimization process.

We trained 2 such GANs on separate sets of scenery datasets, which developed different biases in the 2 trained GANs. These datasets consisted of various landscapes like mountains, rivers, glaciers, deserts etc. We then generated images from both GANs which matched the text "Beautiful landscape" with the highest level of similarity.
We then classified these images into the individual categories i.e. Mountains, rivers, deserts etc. using a CNN trained on these same datasets with the corresponding scenery labels. We used Transfer learning from the ResNET model, to build our CNN, in order to get the highest accuracy of image classification of generated images.
We analysed the number of images of various categories generated by our Clip-GAN model, and which category was displayed the most, the least etc. This shows us the biases in our individual models, which may be that for example, on the first dataset, the trained Clip-GAN model thinks of mountains as more beautiful than rivers, or on the second dataset glaciers are found to be the most beautiful.
Thus we create and train our own GAN models, use the CLIP algorithm to generate images best described by a text string, and check for biases in the models using a CNN model we built.

The caption for every image is the score assigned to every class by the CNN, in the order- [buildings, forests, mountains, glacier, street, sea].

Results:

As we see in the images generated by the BigGAN model, out of 9 images, it assigns mountains, forests, forests, mountains, mountains, street, glacier/mountain(very close values), street and sea respectively to the 9 images.

Thus it finds 3(or 4 if the glacier/mountain case is considered) mountain images, 1 glacier image, 2 forest images, 1 sea image and 2 street images. This tells us that it is biased towards the mountain images, and finds them most beautiful.

Figure 2: This is the DC-GAN built by us, trained for 5 epochs. Since this was taking a lot of training time, and we did not train it for long enough, we do not get very good results. But we see the beginnings of a beautiful scenery as we apply the CLIP-glass algorithm on this GAN model with the above text

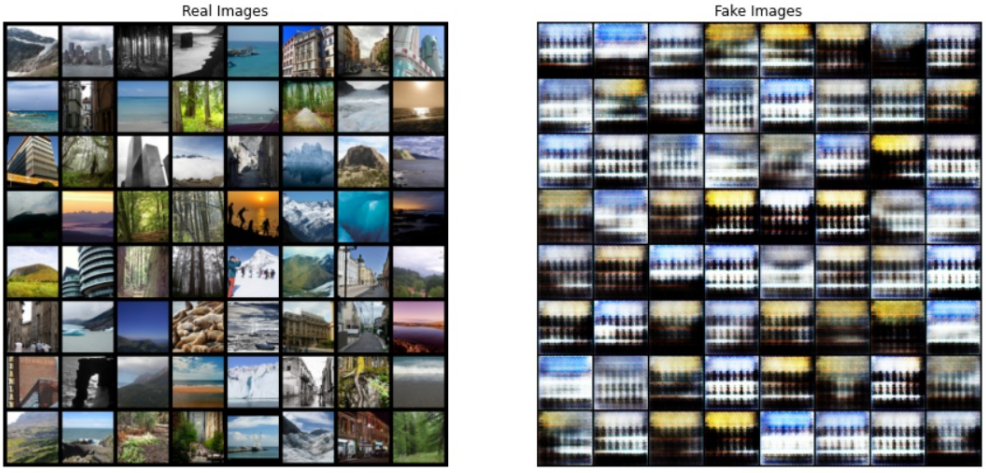

Figure 3: These are the DC-GAN generated images after 200 epochs.

(a) CNN Score = Building: 56.34% (b) CNN Score = Building: 56.76% (c) CNN Score = Building: 45.23%

Figure 4: Images generated by church Style-GAN describing a beautiful scenery. As we see, as the GAN is trained only on church images, all the images generated by it contain churches. We can see that the CNN assigns the highest score to the buildings class for every image, which tell us that since the GAN has only ever seen churches, it thinks that churches are the most beautiful

(a) CNN Score = Mountain: 55.72% (b) CNN Score = Forest: 40.22% (c) CNN Score = Forest: 86.44% (d) CNN Score = Mountain: 55.72%

(e) CNN Score = Mountain: 55.36% (f) CNN Score = Street: 48.11% (g) CNN Score = Forest: 40.58% (h) CNN Score = Street: 82.08%

(i) CNN Score = Sea: 97.3% (j) CNN Score = Glacier: 47% (k) CNN Score = Mountain: 53.98% (l) CNN Score = Forest: 60.79%

Figure 5: Images generated by BigGAN describing a beautiful scenery. We see that the images generated are varied, but BigGAN has a bias towards Mountains(4 images) and Forests(4 images) for 12 images as compared to other labels.

References

[1] Federico Galatolo., Mario Cimino., and Gigliola Vaglini. Generating images from caption and vice versa via clip-guided generative latent space search. *Proceedings of the International Conference on Image Processing and Vision Engineering*, 2021.

[2] Sieun Park. Key concepts of biggan: Training and assessing large-scale image generation, Mar 2021.

[3] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[4] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2015.