Deep Fake Image Generation Using Deep Convolutional Generative Adversarial Network

GEORGE MASON UNIVERSITY

Team Members: Maazuddin Mohammed, Nina Nnamani, Rutuja Shinde

As the crime rates have increased due to fake images and videos, it has become the need of the hour today to build technologies that could identify these threats and protect us from any potential scams. This project aims to implement a DCGAN (Deep Convolutional Generative Adversarial Network) to generate "fake" images based on an existing dataset. The generated images can then be used to build a classifier which can identify an image as real vs fake. GANs like a DCGAN have been used widely to create Deep Fakes. We aim to use DCGAN to build high-quality deep fakes.

Deep Fakes are the altered videos that appear to imitate a real person through techniques such as swapping facial features in order to make the person in the video appear as though it is an original video. They are videos or bots which give us the feel of a real person talking when in reality they are “fake”. Deep Fake videos of politicians and celebrities are being created increasingly to mislead the people. A very famous example of a Deep Fake video is of the late actor Paul Walker where they used GAN to create an artificial version of him in the last scene of the movie Fast and Furious7.

Dataset

The **CIFAR-10** dataset is described as a widely used dataset for machine learning computer vision tasks such as image classification, object recognition, etc. The dataset contains 60,000 colored images of 32 by 32 size across 10 classes that are either animals or transportation means. There are 6000 images for each of the 10 classes. The dataset is divided into a training set and a test set with 50,000 and 10,000 images, respectively. The training set is further divided into five batches and one test batch, where there are 10,000 images each. Of note, the test batch’ selection was randomized to include 1,000 images per class, while the training batch consists of the latter randomized ordered images with 5,000 images per class. We have subset the **CIFAR-10** dataset and use only **DOG** class to implement a DCGAN to generate high-quality fake images.

Figure 1: Original CIFAR-10 dataset for the DOG class

What Are GAN’S?

Generative Adversarial Networks also known as GANs are a set of deep-learning based adversarial models. That means they are able to generate new content from the given content. Generative modelling is a type of unsupervised learning algorithm in machine learning that learns and discovers patterns or irregularities in the input data in such a way that the model generates an output of new data instances that look almost like the input data. Long cut short a GAN takes in input data identifies its patter and underlying probability distribution and generates or mimics it to create fake versions of the original data and then classifies if the generated data is fake or real. For example a GAN can create images that look like photographs of human faces even though the face doesn’t belong to any real person. A GAN model architecture involves two sub-models namely : a Generator model that creates the new instances(data) and a Discriminator model that classifies if the generator created data is real or fake. The generator tries to fool the discriminator, and the discriminator tries to keep from being fooled.

Generator - The generator plays the part of a falsifier and attempts to make music/image/speech from random noise. It figures out how to plan from an inert space to a specific data distribution of interest. It for the most part actualizes a Deconvolutional Network to do as such.

Discriminator-The Discriminator then again plays the function of the evaluator and attempts to recognize the fake data (made by the Generator) from the genuine one. It is generally executed as a Convolutional Network.

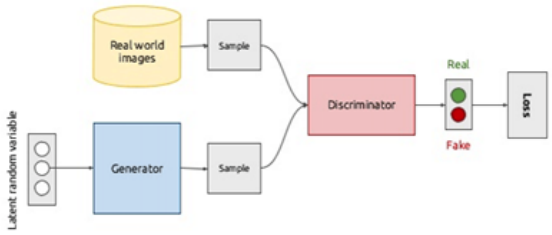
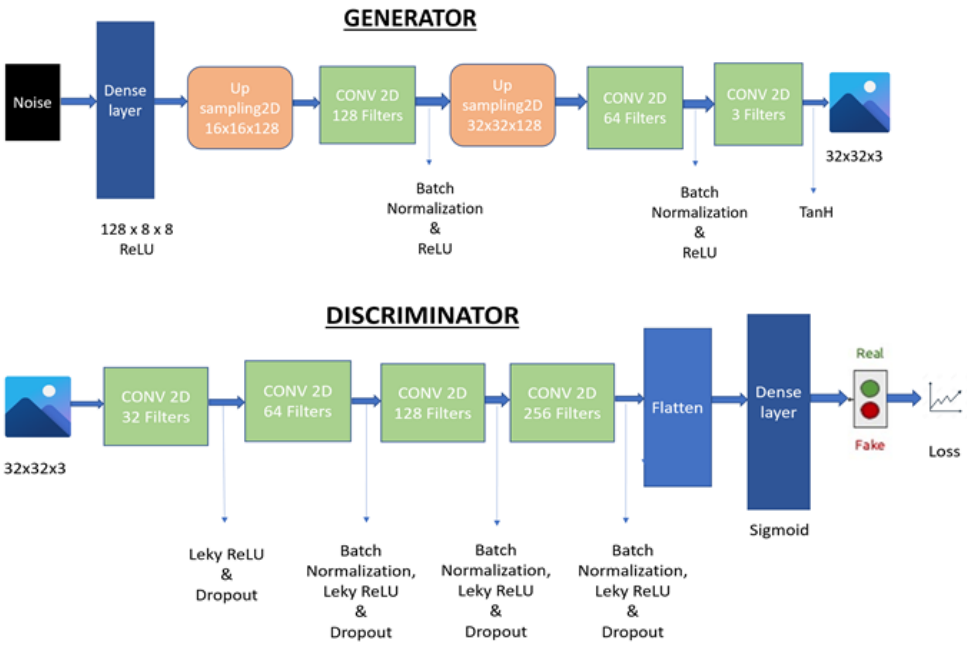


Figure 2: GAN Architecture

Literature review

There are GAN models available which generates fake dog images using the Stanford dog dataset which has almost 20,000 images of different breed of dogs (8). There was a famous Kaggle competition for selecting the best GAN model using the Stanford dog dataset (9). In this project a DCGAN model is made using the DOG class in CIFAR-10 dataset which has almost 5000 images. The CIFAR-10 dataset has 10 classes in total. All the previous attempts use all the 10 classes for their classification or image generation. None of the previous attempts have used the 'DOG' class from the dataset to generate fake images using a GAN and classify it. We have built 3 models, one of which is our baseline model for our comparison.

DCGAN Architecture



The baseline model comprises of a simple DCGAN architecture, which uses Keras Sequential API to create the model for both the generator and discriminator. The shape of the input images for the DOG dataset is 32x32. Batch normalization has been performed for all the layers for both the Generator and Discriminator. The layers in Generator uses ReLU activation except for the output layer which uses Tanh activation function since the images are normalized between -1 and 1. The layers in Discriminator uses Leaky ReLU activation except for the output layer which uses Sigmoid activation function. A convolutional neural network have been used to build the Generator and the Discriminator. The Discriminator has (0.3) dropout layers in its network. Binary cross-entropy is used to calculate the loss in the discriminator as it is a binary classification problem. Adam's Optimizer to compile the discriminator for improved learning. The model was trained with 800 epochs with a display interval for every 50 epochs. A batch size of 32 images was given to the generator for training. The discriminator and generator losses are plotted. The Generator generates fake images of the DOG class from random noise and gives it to the discriminator to identify which is fake and real. As the computational requirement to train a DCGAN is high we used the GPU on the Goggle Collab and stored the data on google drive.

DCGAN with different learning rates

The baseline DCGAN model was tuned by changing the learning parameters of the ADAM's optimizer. We tried 2 models with different learning rates of 0.0002 and 0.0003 for the optimizer. We transposed the layers for both the Generator models. The models with a learning rates of 0.0002 and 0.0003 were trained for 800 epochs. The problem of dying ReLU arises in the baseline model. To address this issue Leaky ReLU is used as the activation function for both these models to improve their learning rates.

Best Model & Comparison

The tuned DCGAN model with a learning rate of 0.0003 performed better than both the baseline and the other tuned model. The Generator given the input images performed a good enough task of creating comparatively clear images for the CIFAR-10 (DOG class) dataset. The loss function shows how much has the Generator deviated from the distribution of the real data. Ideally, the loss for both the generator and discriminator should be as low as possible indicating the generator created good enough images from noise by replicating the original images and also that the discriminator has classified the images correctly. The loss however for the baseline DCGAN model varies drastically indicating it did not perform well in generating the images. The resultant images for all the 3 models are shown below with a step by step comparison of the images formed at each epoch interval for all 3 models.

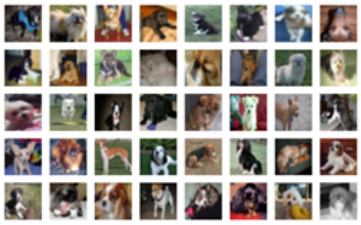
Results

---

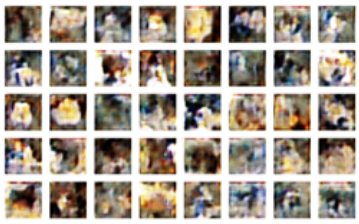
1. DCGAN

Figure 3: Training Loss for Discriminator and Generator

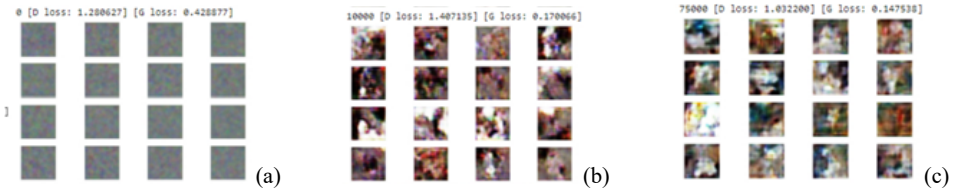
This graph shows the loss for both the generator and discriminator of the baseline DCGAN model.



Original Training dataset images



GAN generated images

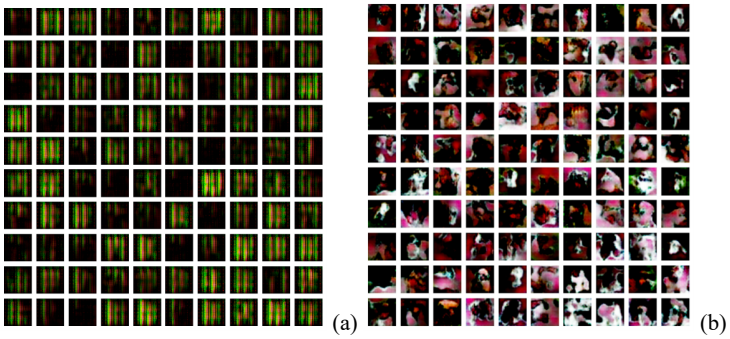


(a) (b) (c)

Figure 4: GAN generated images for Baseline Model (a). Noise (b). 100 Epoch (c). 800 Epochs

---

2. DCGAN with learning rate = 0.0002



(a) (b)

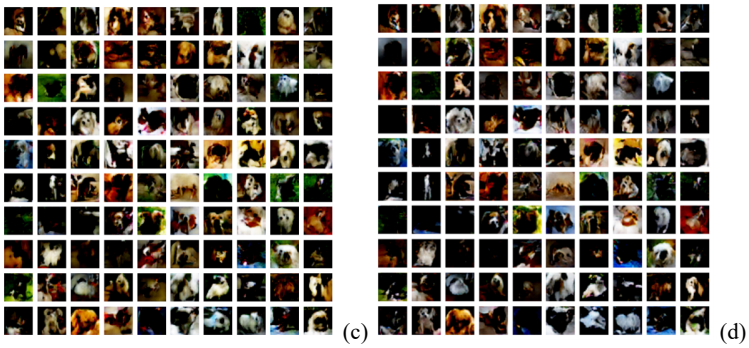


Figure 5: GAN generated images with learning rate = 0.0002 (a). Noise (b). 300 Epoch (c). 500 Epochs (d). 800 Epoch

---

3. DCGAN with learning rate = 0.0003

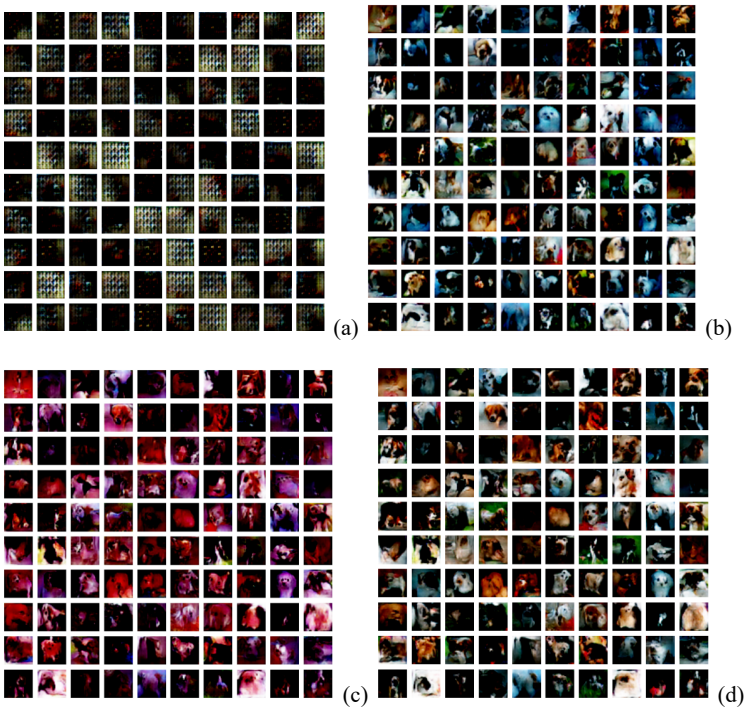


Figure 6: GAN generated images with learning rate = 0.0003 (a). Noise (b). 300 Epoch (c). 500 Epoch (d). 800 Epoch

Conclusion

GAN's are state of the art technology studied widely today. Deep fakes created using a GAN are studied and implemented for various purposes. One of the main purpose is to understand the underlying pattern and distribution of data by training the model adversarially. This project was focused on building and implementing a DCGAN which is a modified version of GAN that uses deep convolutional neural networks with dropout layers to create fake images from the original dataset. The model that performed better than other models for generating fake DOG images used a learning rate of 0.0003, with transposed layers in the generator, a Leaky ReLU activation function, and an Adam's optimizer.

Future Work

This project scope can be widened and applied for multi-class classification by using a Conditional GAN. In Conditional GAN labels are also provided as the input to the Generator and the Discriminator during training. All the 10 classes in the CIFAR-10 data set can be used as the training data and the model would generate random fake images that mimic any of these categories, the discriminator will then classify the real and fake images for each of these 10 classes and calculate loss as a measure of accuracy. Such GAN's can also be used for generating and identifying deep fake videos which is a hot research topic today. More versions of this GAN can be used in the future by hyperparameter tuning of the existing ones or making changes such as adding noise to the labels before feeding them to the discriminator, sampling from a Gaussian distribution rather than from a Uniform Distribution, generate mini-batches for real images and fake images separately, using a pre-trained discriminator or a generator, apply transfer learning on the dataset and adding some gaussian noise to the images before feeding it to the Discriminator.

References

1. Sharma, S. (2019, February 14). Activation Functions in Neural Networks. Retrieved October 15, 2020, from https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

2. Anwar, A. (2020, November 06). What is Transposed Convolutional Layer? Retrieved November 01, 2020, from https://towardsdatascience.com/what-is-transposed-convolutional-layer-40e5e6e31c11

3. IBM Developer Advocate in Silicon Valley. (2019, December 26). Deepfakes and the world of Generative Adversarial Networks. Retrieved October 12, 2020, from https://medium.com/@lennartfr/deepfakes-and-the-world-of-generative-adversarial-networks-bf6937e70637

4. Hui, J. (2020, September 30). Detect AI-generated Images & Deepfakes (Part 1). Retrieved October 12, 2020, from https://medium.com/@jonathan_hui/detect-ai-generated-images-deepfakes-part-1-b518ed5075f4

5. “The CIFAR-10 dataset” https://www.cs.toronto.edu/~kriz/cifar.html

6. K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng and F. Wang, "Generative adversarial networks: introduction and outlook," in IEEE/CAA Journal of Automatica Sinica, vol. 4, no. 4, pp. 588-598, 2017, doi: 10.1109/JAS.2017.7510583.

7. C. Shorten, “Deeper into DCGANs,” Medium, 09-May-2019. [Online]. Available: https://towardsdatascience.com/deeper-into-dcgans-2556dbd0baac.

8. Stanford Dogs dataset for Fine-Grained Visual Categorization. [Online]. Available: http://vision.stanford.edu/aditya86/ImageNetDogs/.

9. “Generative Dog Images,” Kaggle. [Online]. Available: https://www.kaggle.com/c/generative-dog-images.