Synthetic Card Images Genereation

Abdurahman Fayad ABDURAHMAN.FAYAD@GMAIL.COM
Gaia Sandra Schillaci GAIASCHILLACI1999@GMAIL.COM

1. Model Description

The type of the model is Conditional DCGAN (Deep Convolutional Generative Adversarial Network), it is an extension of traditional DCGANs that introduce conditional information to the generator and discriminator networks. In a Conditional DCGAN, the generator takes additional input, such as class labels or other conditional information, along with the random noise vector, to generate more controlled and specific outputs. This conditioning allows the generator to generate samples conditioned on specific classes or attributes, resulting in more targeted and diverse output generation. The discriminator network is also modified to consider the conditional information when making its discrimination decisions.

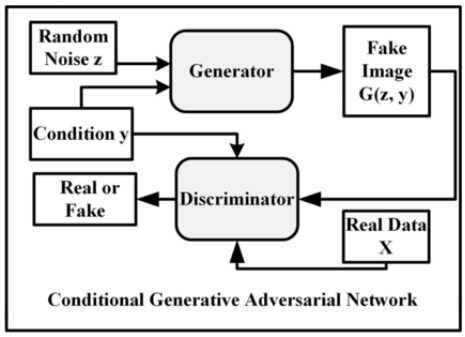

Figure 1: Conditional GAN Architecture

The goal of the model that is able to synthesize images at a resolution of (64x64) and suitably conditioned through image labels. Once the model is trained, its evaluation is carried out in terms of Inception Score and FID. The model will be used to generate synthetic dataset as a form of data augmentation for training a classifier on the labels used for conditioning and assess whether it provides a contribution to classification performance or not.

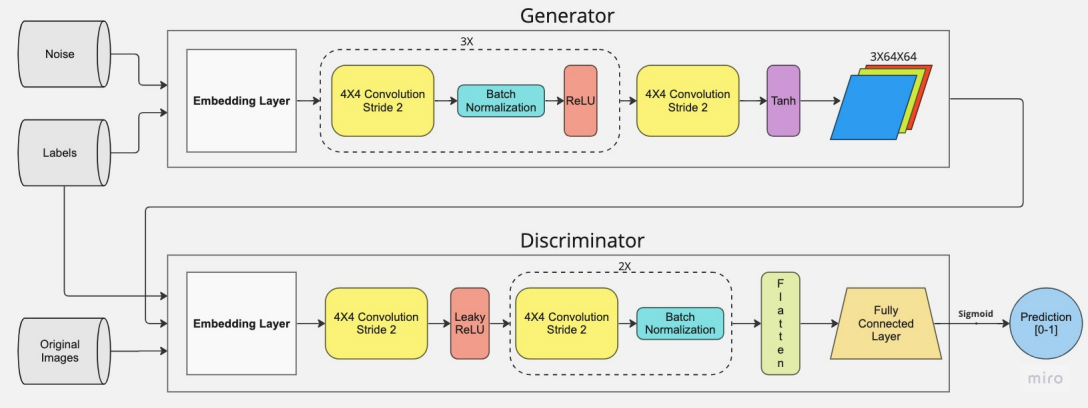

Figure 2: Generator and Discriminator Models

2. Dataset

The dataset used for the homework consists of playing card images. All images are 224 x 224 x 3 and have been cropped so that only the image of a single card is present and the card occupies well over 50% of the pixels in the image. There are 7624 training images, 265 test images and 265 validation images. The train, test and validation directories are partitioned into 53 sub directories, one for each of the 53 types of cards.

Figure 3: Example images from dataset

3. Training procedure

Hyperparameters and Optimizer

In the training the data have been grouped in batches of 64 elements, shuffled for each of the 300 epochs that have been performed, while setting 2 as number of workers. All images have been resized to 64X64.

The optimizer used for both generator and discriminator is Adam, with learning rate=0.0001, beta1=0.5 and beta2=0.999. The momentum term (beta1) of 0.5 allows a faster convergence during the initial iterations, while the second-order moment term (beta2) of 0.999 helps to stabilize the update steps and to reduce the effect of noisy gradients.

Loss

The loss used is the Binary Cross Entropy(BCE) for both generator and discriminator, it can be represented as:

$$BCE = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i)$$

Figure 4: Binary Cross Entropy

Where y_i represents the ground truth value, and $\hat{y}_i$ represents the output value.

Quality Metrics

The Inception Score (IS) is an algorithm used to assess the quality of images created by a generative image model, the score is calculated based on the output of a separate, pretrained Inceptionv3 image classification model applied to a sample of images generated by the generative model.

It has been somewhat superseded by the related Fréchet inception distance. While the Inception Score only evaluates the distribution of generated images, the FID compares the distribution of generated images with the distribution of a set of real images ("ground truth").

Both FID and Inception score has been calculated every **30** epochs using modules imported from Pytorch library.

4. Experimental Results

GAN Model Evaluation

As discussed earlier, the model is evaluated by calculated the inception score, as well as the FID, however, it is important also to visualize the generated images and assess their quality by us, below we see a comparison between a batch of real images and a generated one.

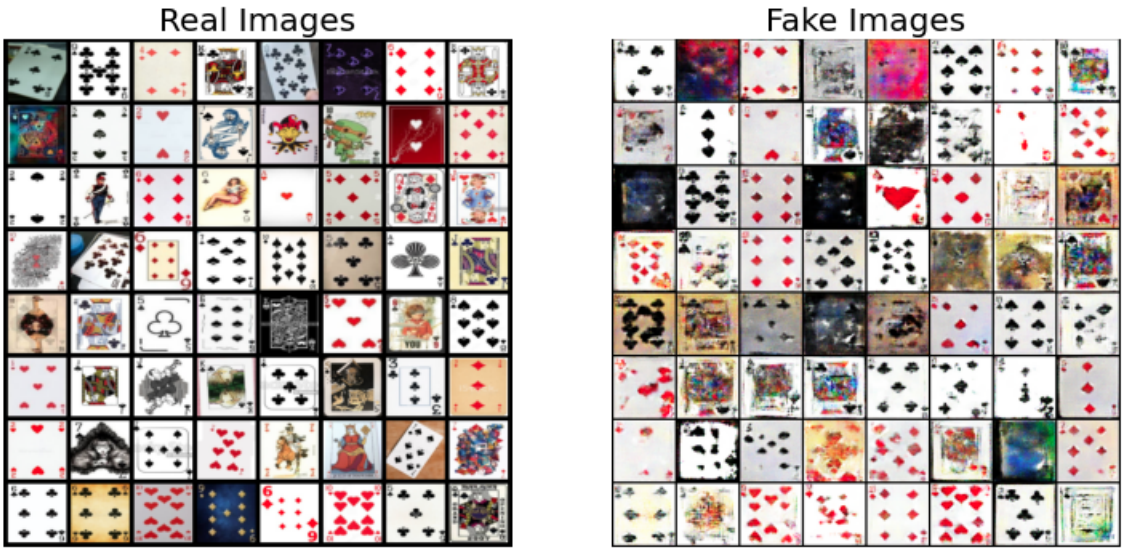

Figure 5: Visual Comparison

As we can see, the generated images has a big variety between good ones, average, and very noisy.

Next we plot the changes in FID and Inception Score(Mean and Standard deviation). We can see below that the FID is almost stable at around 100 after 200 epochs, as well as the mean in the inception score, however the standard deviation is not stable.

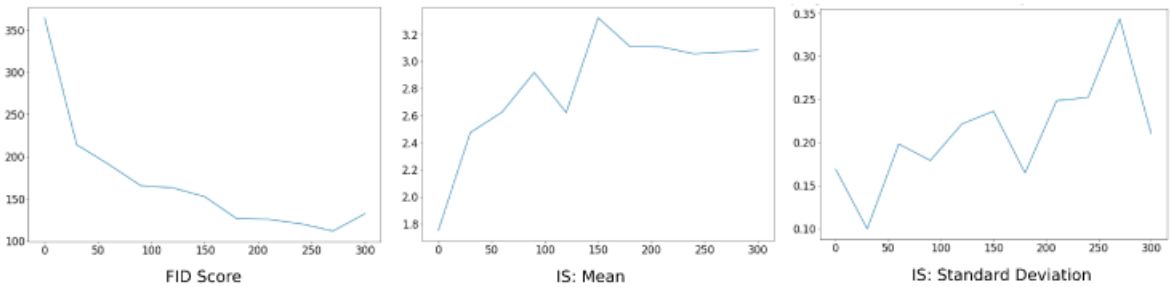

Figure 6: FID and Inception Score

Classification Evaluation

Now we will use the generated synthetic dataset as a form of data augmentation for training a classifier, the classifier model is a ResNet-18 as pre-trained weight set to None, with SGD Optimizer, Cross Entropy Loss, Learning rate=0.001 and Momentum=0.9 . The architecture of Res-Net-18 is as follows:

Layer Name	Output Size	ResNet-18
conv1	$112 \times 112 \times 64$	7×7, 64, stride 2
conv2_x	$56 \times 56 \times 64$	3×3 max pool, stride 2
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$
conv3_x	$28 \times 28 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
conv4_x	$14 \times 14 \times 256$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$
conv5_x	$7 \times 7 \times 512$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$
average pool	$1 \times 1 \times 512$	7×7 average pool
fully connected	1000	512×1000 fully connections
softmax	1000	

Figure 7: ResNet-18

As we can see in the below figure, the accuracy and the loss had a slight improvement on the test dataset, after adding the generated images to the original training dataset as a form of data augmentation.

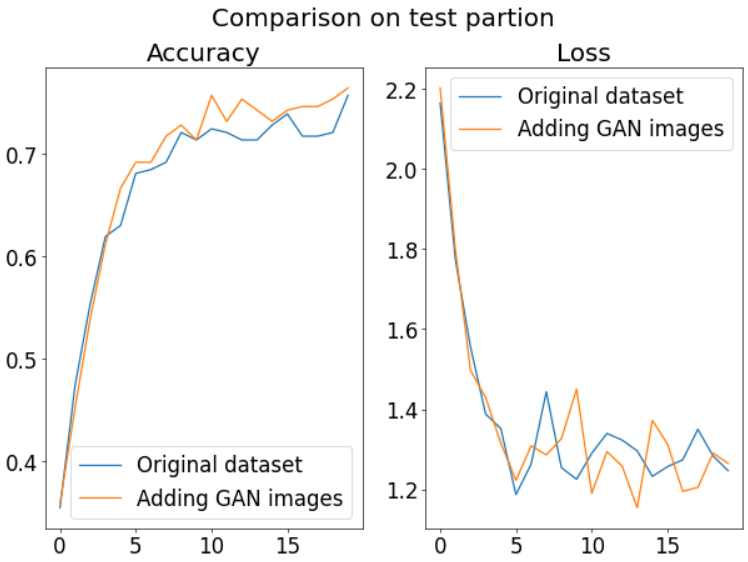

Figure 8: Accuracy and Loss Comparison