# Question 1

## Input Vector (50)

A group of squares with text

Description automatically generated

### Vector size is 50 and increase to 10000 epochs with 128 batch size

When the size of the latent space was changed from 100 to 50, we observed considerable effects in terms of quality and variability of the generated images. When the latent space was reduced to 50 the outputs were of less detail and were also less various, as the generator seemed unable to work with such a complex data distribution. The images generated seemed to be more of a blur with not clearly definable objects in pictures. However, in place of preserving the latent space size of 100, it made a better compromise, in that, the generator was able to produce better and more diverse images without upsetting the training dynamics. For that reason, when a latent space of 100 was maintained the loss of images of good quality was lower when compared to when a latent space of 50 was used.

## Input Vector (100)

A group of squares with text

Description automatically generated

### Vector size is 100 and increase to 10000 epochs with 128 batch size

The training was sustained for a period of 10,000 epochs after which some observations were noted about the deviations in the discriminator accuracy, which worsened to as low as 0.117% at epoch 3,000, which meant that the generator was starting to provide hard images for the discriminator to classify. It should also be noted that the loss curves of both the discriminator and the generator were high and non-indicative, which is a common characteristic in training GANs. Nonetheless, one could observe how the quality of images improved progressively over the epochs, in a GIF or a slideshow provided alongside, meaning a certain degree of sophistication was ingrained in the generator.

## Different optimizers

A group of squares with black text

Description automatically generated

### Changed to RMSprop optimizer

Shifting the optimizer from Adam to RMSprop or SGD, for both the generator and the discriminator in the GAN architecture, brings out significant changes in training performance as well as image quality. The use of RMSprop usually gives training dynamics which are less prone to loss fluctuations, and this may also be useful in equalizing the generator and discriminator so as to avoid one dominating the other. However, the other side of this kind of stability is the possibility of slower rates of improvements and more epochs being needed to achieve comparable image quality when using Adam. Alternatively, leaping to SGD may even slow the process down more because this optimizer has a constant learning rate, no momentum and is mostly suboptimal since it requires fine tuning with schedules. This means that both RMSprop and to a greater extent SGD although offering stability in training span several epochs getting quality images similar to that of Adam’s usage.

## Different Batch sizes

## A group of squares with text Description automatically generated

### Change to 64 batch\_size

The tests performed investigated the effect that varying the size of individual batches had on the efficacy and stability of GAN training. Using a batch size of 64 and 50 respectively, results showed that the smaller batch size (50) increased the accuracy of the Discriminator up to 81.64% from 52.34% on the larger size. The loss of the Generator, however, increased dramatically with the smaller batch size portraying instability in the training process. These results further imply that smaller batch sizes improve the operation of the Discriminator but on the other hand strain the Generator’s performance thus the need for optimization to harmonize both models.

## Conclusion

Direct changes in parameters like the size of the latent space, the choice of optimizers, and the batch sizes influenced the quality and variety of images that were produced heavily. There was a striking decline in the details of the pictures generated when the latent space size decreased from 100 to 50, and the generator also failed at drawing out any intricate patterns. A combination of a latent space size of ten thousand, a reasonable batch size of one hundred and twenty-eight and running the network for a thousand epochs gave better results of clearer and more diverse images with the generator progressively learning how to make better images. The use of RMSprop improved the quality of training by tempering the variability in the lost values because although this improved the training of the model, the phenomenon prolonged the number of epochs to ach typical image quality. In contrast, the rapid convergence observed among others because of Adam was offset by undesired behavior at some parts of the training process. The 128 batch size was also adequate since it allowed for easy updates during training leading to good gradual quality and diversity of images over the number of epochs. To sum up, an efficient combination of the high latent spaces provided on demand and the optimizers and batch sizes led to the better quality and diversity of images.

### Note

"Answers to Questions 1, 2, and 3 in the notebooks."