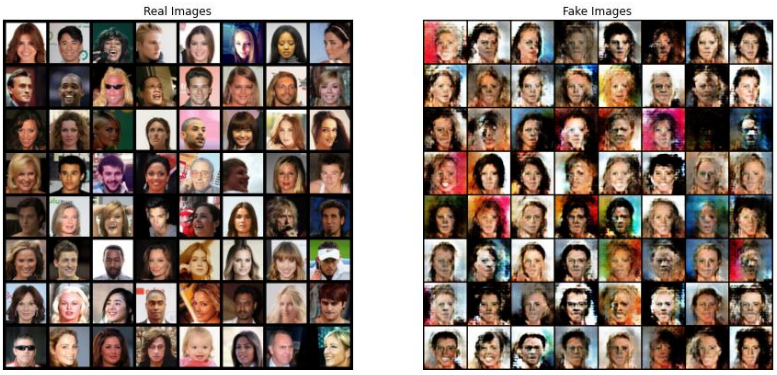
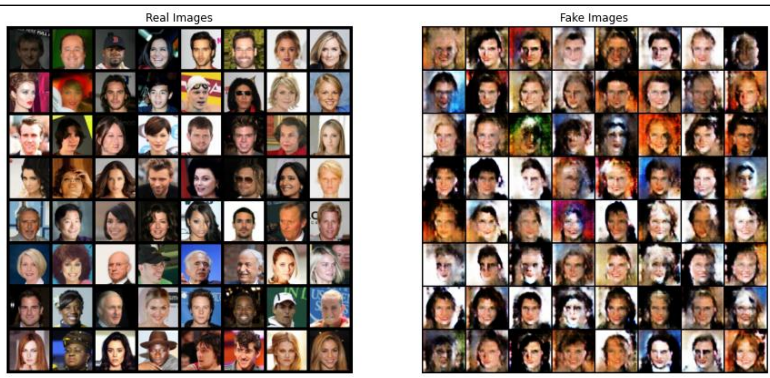
Question 1: Change the random seed to generate different visualizations, analyze the impact of random seeds on visualization outcomes and evaluate the robustness of DCGANs.

We first set the parameters num_epochs = 1, lr = 0.0002, batch_size = 128, keeping these parameters the same, we set the Random Seed to 728, 999, and 1, and observe the differences in the images generated.

Random Seed = 728		**Average G_loss 5.19535 Average D_loss 0.64196**
Random Seed = 999		**Average G_loss 5.17443 Average D_loss 0.66090**
Random Seed = 1		**Average G_loss 5.18482 Average D_loss 0.66976**

From the comparison above, we can draw the following conclusion: The offsets for both G and D are within 0.02, thus, the GAN is robust to changes in the random seed.

Question 2: Modify the number of epochs and discuss how results change in different epoch settings.

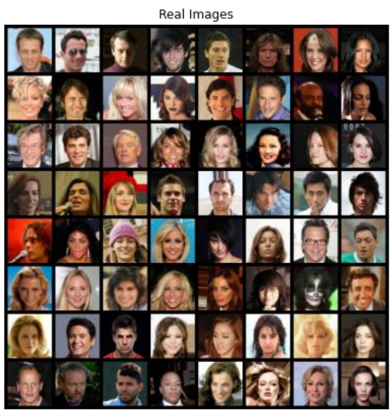
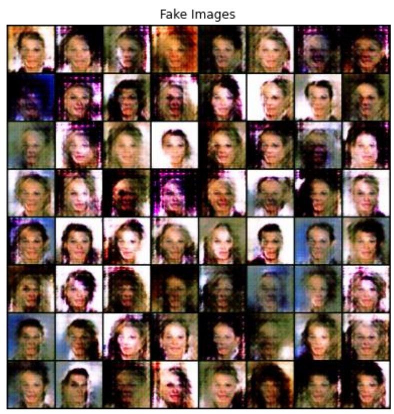
We first set the parameters manual_Seed = 999, lr = 0.0002, batch_size = 128. Under these consistent conditions, we observe the changes in the data at epochs = 1, 2, and 3.

Epochs =1			**Average G_loss 5.19535 Average D_loss 0.64196**
Epochs =2			**Average G_loss 4.29896 Average D_loss 0.67143**
Epochs =3			**Average G_loss 3.85396 Average D_loss 0.68477**

From the results above, we can see that as the epochs increase, G's loss gradually decreases, while D's loss tends to increase. However, from the images, it is evident that the quality of the generated pictures becomes more refined, but they do not accurately reproduce the original details of the images (such as hair, clothing, etc.) Since the objective of the GAN loss function is to minimize G's loss and maximize D's loss, from the perspective of the loss function, the model's error decreases as the number of epochs increases.

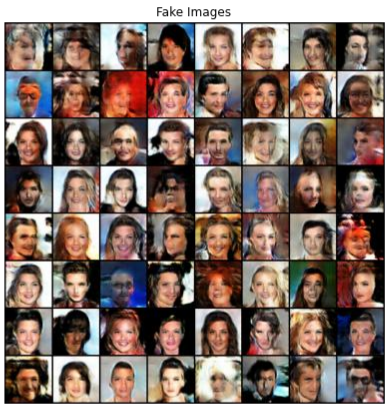
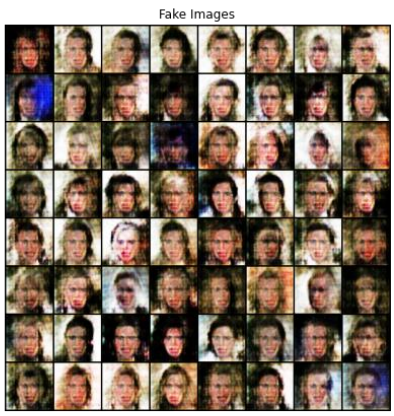
Question 3: Change the learning rate and batch size and discuss the effects of these hyperparameters on the training process.

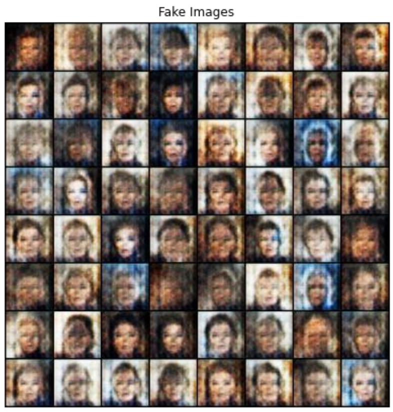
First, we explore the impact of the learning rate on training, maintaining model parameters manualSeed = 999, num_epochs = 2, batch_size = 256, while adjusting the lr values to 0.0001, 0.0005, 0.001 and observing the results, as shown below:

Learning rate 0.0001	 	**Average G_loss 8.63111 Average D_loss 0.28469**
Learning rate 0.0005	 	**Average G_loss 3.24576 Average D_loss 1.16500**
Learning rate 0.001	 	**Average G_loss 3.16951 Average D_loss 1.10190**

From the figure above, we can see that when the learning rate is set to 0.0001, the images remain relatively blurry after iterative training, particularly with facial details being difficult to discern. Essentially, all images exhibit similar features with little variation. This may be attributed to the small learning rate causing minimal parameter updates, as reflected by the average values of the loss function. On the other hand, when the learning rate is set to 0.001, although the loss function indicates smaller errors compared to a rate of 0.0005, the generated images incorporate many unnecessary noise points, such as pink dots in the background. Therefore, overall, setting the learning rate to 0.0005 yields the best performance of the model.

Next, we explore how changes in batch size affect the generation of images. In this case, we assume manual_Seed = 999, num_epochs = 2, and lr = 0.0003, with batch sizes of 128, 256, 512 , 1024for variation. The results are as follows:

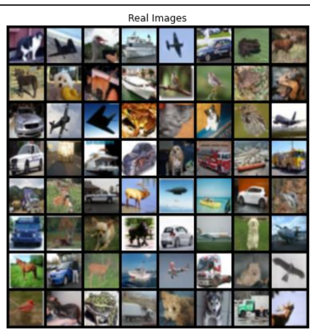
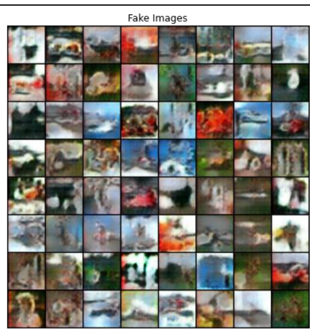
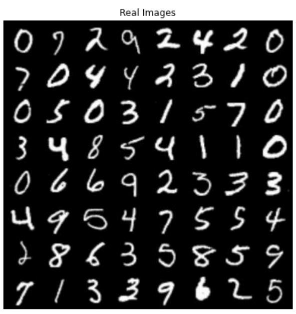
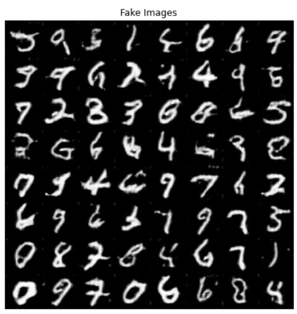
Batch Size 128	 	**Average G_loss 3.77979 Average D_loss 0.93408**
Batch Size 256	 	**Average G_loss 4.24599 Average D_loss 0.89057**
Batch Size 512	 	**Average G_loss 4.63432 Average D_loss 0.99142**
Batch Size 1024		**Average G_loss 6.43754**

	 		**Average D_loss 1.01046**

From the figure above, we can see that as the batch size increases, particularly when it reaches 512 and 1024, the loss function significantly increases. This indicates that the model's bias also grows with the increase in batch size. This can be observed in the generated images as well; as the batch size increases, the details of the facial features and skin texture in the generated images deviate more from the original data. Therefore, we can conclude that the larger the batch size, the less precise the images generated by the GAN become.

Bonus Question: Use the CIFAR-10 dataset to train the DCGAN model, analyze the results, and compare the performance of DCGAN on the Celeb-A Faces dataset and the MNIST dataset.

Compile the results of the three datasets into the table below, as shown in the following figure.

CIFAR-10	**manualSeed 999 batch_size 128 num_epochs 2 Learn Rate 0.00075**			**Average G_loss 5.44961 Average D_loss 1.05899**
MINIST	**manualSeed 999 batch_size 128 num_epochs 2 Learn Rate 0.00075**			**Average G_loss 2.44961 Average D_loss 1.50989**

Celeb-A Faces	**manualSeed 999 batch_size 128 num_epochs 2 Learn Rate 0.00075**	 	**Average G_loss 2.53015 Average D_loss 1.21244**

Considering the images and the results of the loss function, we can roughly assume that, in terms of the DCGAN model, the performance on the MINIST and Celeb-A Faces datasets is superior to that on CIFAR-10. This is because the basic shapes of objects are not discernible in CIFAR-10, while they are apparent in the other two datasets. However, for the MINIST dataset, although the shapes generated are somewhat similar due to its simplicity, the numbers generated are not accurate (for example, a 0 is generated as a 7), indicating that there is room for improvement in the model’ s accuracy.