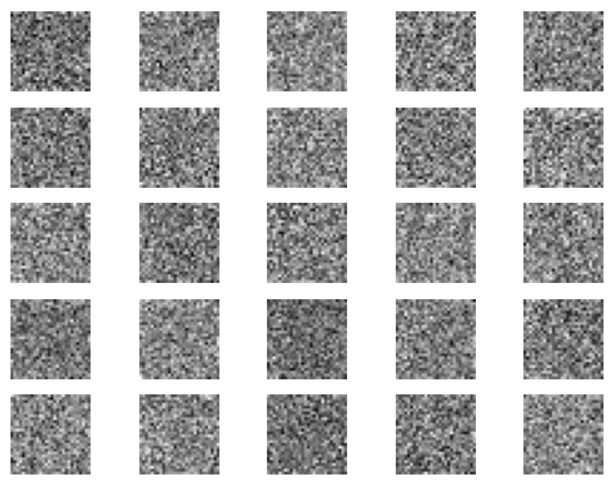
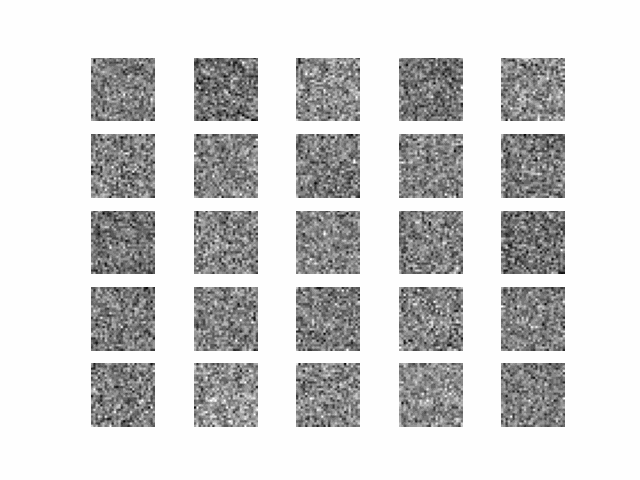
Part 1

1. Noise vector size 50 Noise vector size 100

A group of white dots

Description automatically generated with medium confidence



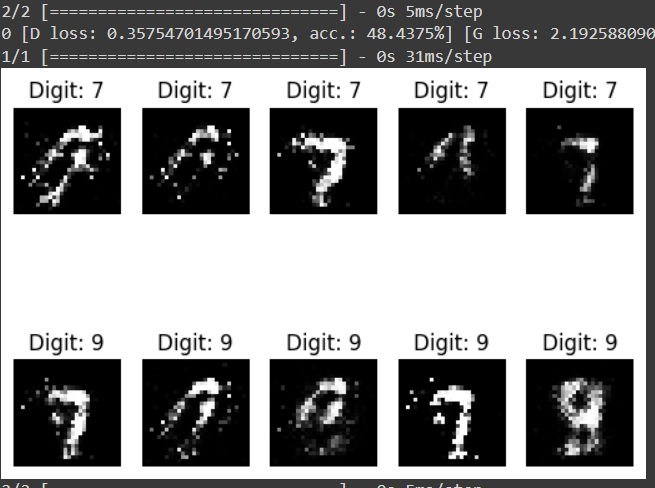
1. Three different batch sizes (32, 64, and 128) were used during GAN training, and the results showed significant differences in training performance,

**Batch Size 64**: Provided the best balance between stability and performance. The discriminator loss was 0.685, and accuracy was 60.16%, indicating a reasonably effective discriminator.

**Batch Size 32**: The discriminator had a high loss of 1.383 and very low accuracy of 9.38%, suggesting that the smaller batch size introduced too much noise into the training process, leading to instability.

**Batch Size 128**: Initially resulted in poor performance, with a high discriminator loss of 3.69 and low accuracy of 3.67%. Larger batch sizes tend to stabilize training over time, but may start off less effective due to slower updates.

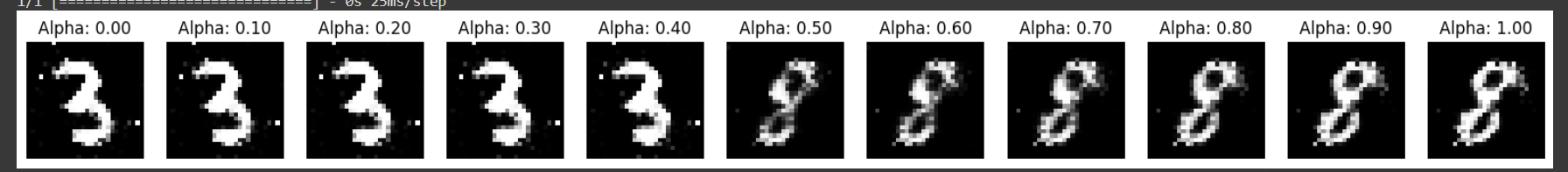
Part 2



2)

Label smoothing was implemented by replacing the real labels of 1 with random values between 0.9 and 1 during training, which encouraged the discriminator to be less confident in its predictions. This technique improved the stability of the training process, allowing the GAN to converge more effectively and reducing the risk of overfitting. As a result, the generator produced higher-quality images with greater diversity. Overall, label smoothing enhanced the training dynamics, leading to a more robust model capable of generating more realistic outputs.

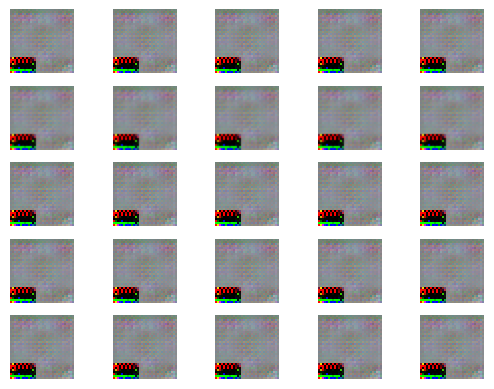
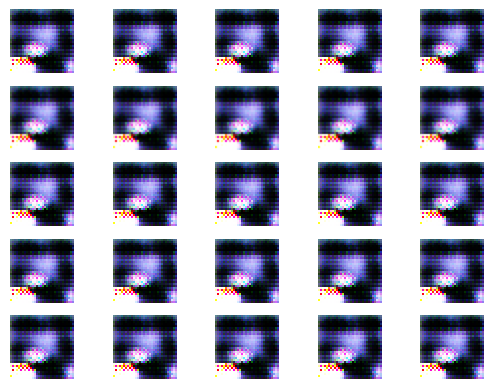
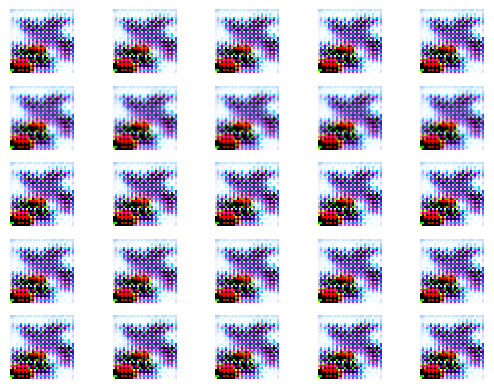
3)



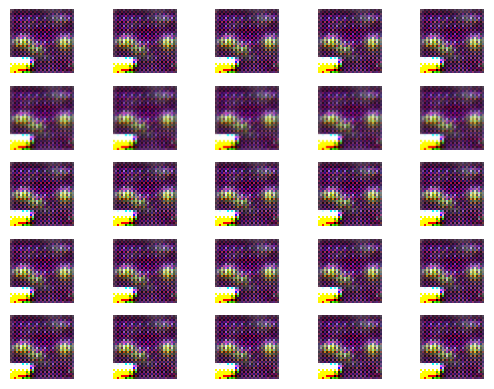
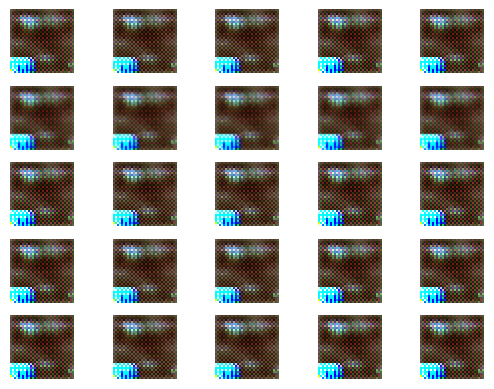
Part 3

1. Generating automobile images,

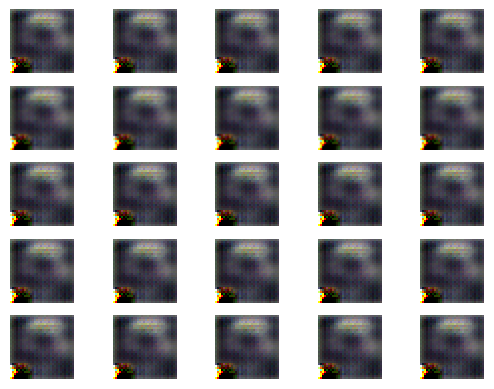
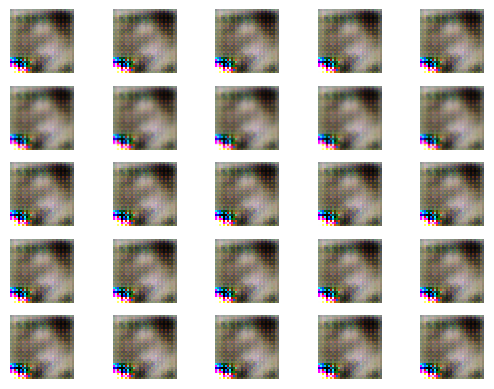
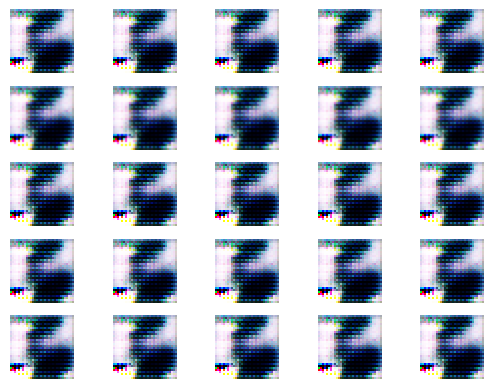
epoch\_0 epoch\_100 epoch\_200



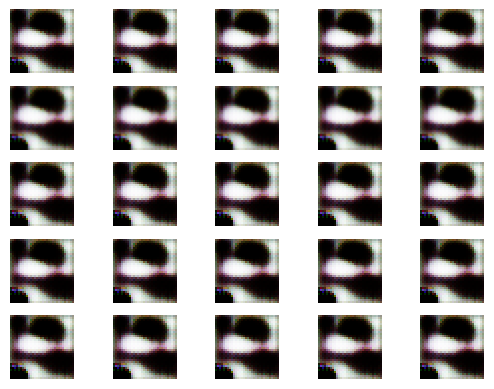
epoch\_300 epoch\_400 epoch\_500



epoch\_600 epoch\_700 epoch\_800

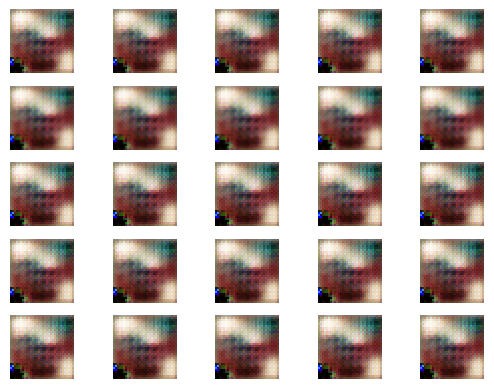
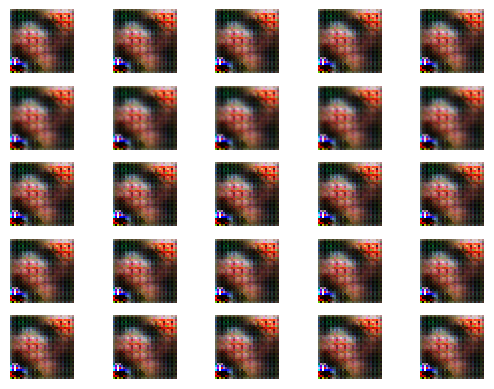
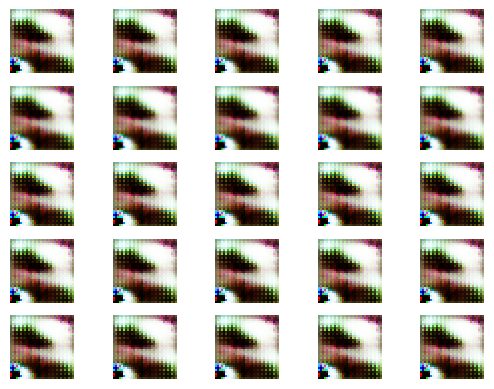


epoch\_900

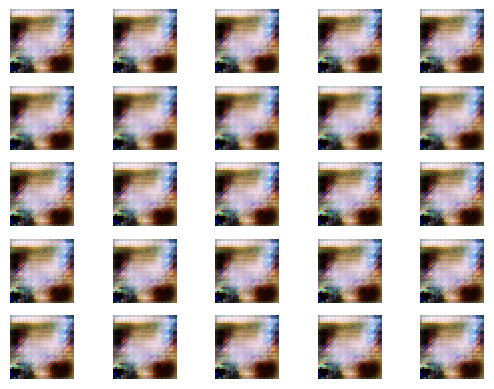
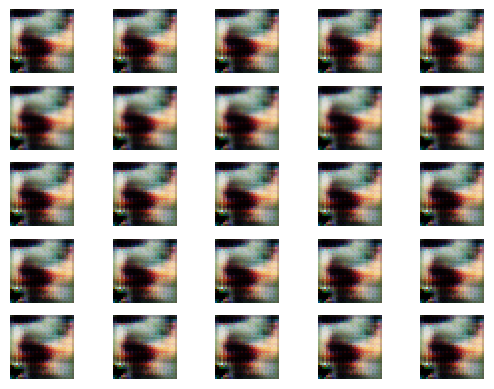


1. After augmentation

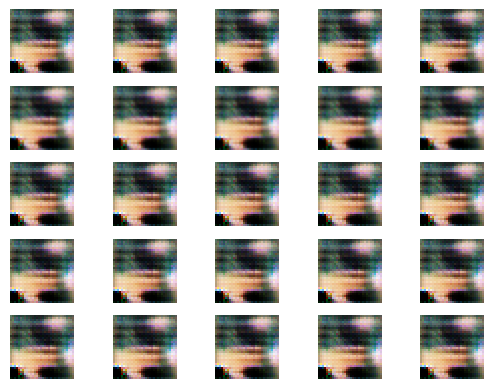
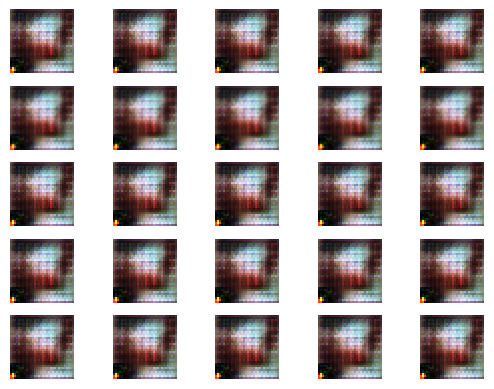
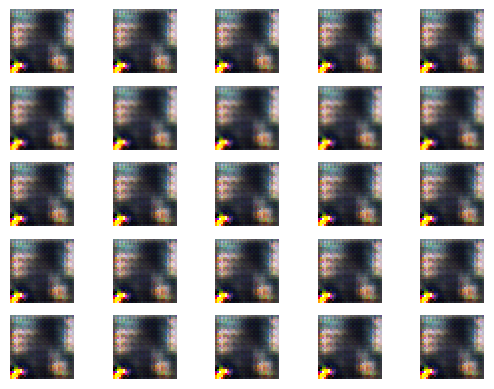
epoch\_0 epoch\_100 epoch\_200



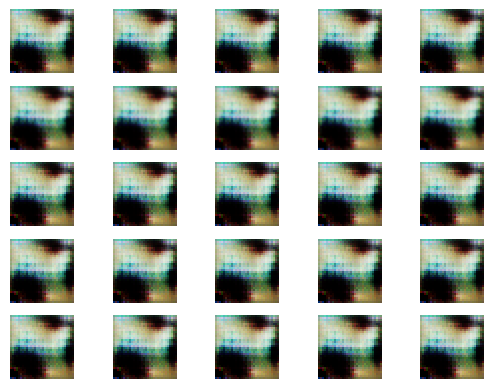
epoch\_300 epoch\_400 epoch\_500



epoch\_600 epoch\_700 epoch\_800



epoch\_900

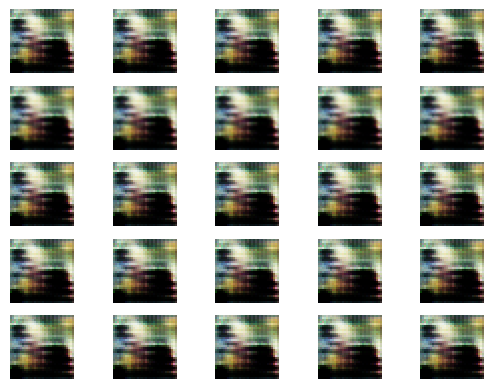
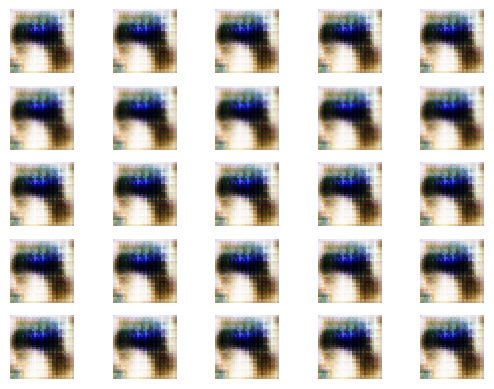


3)

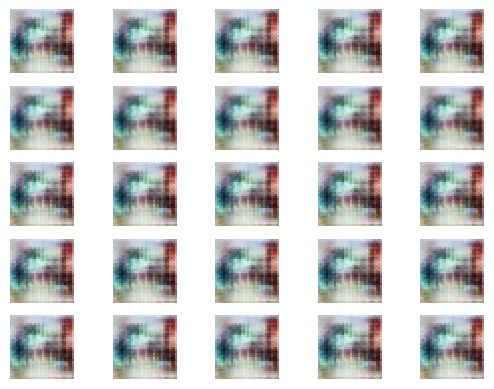
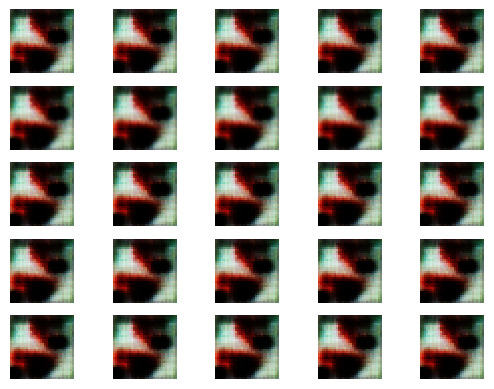
Inception Score at epoch 2000: 1.0000008344650269 ± 2.77017363714549e-07

Inception Score at epoch 5000: 1.0 ± 6.529361940010858e-08

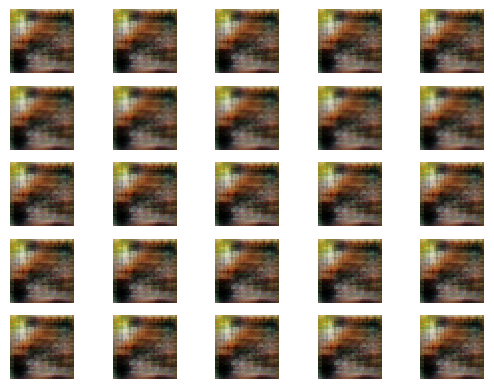
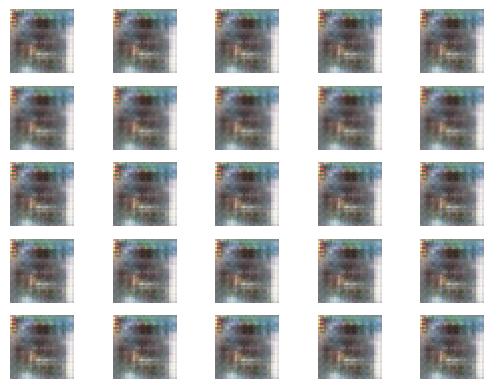
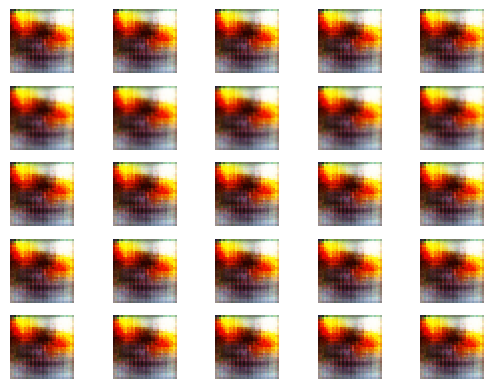
epoch\_0 epoch\_1000 epoch\_2000



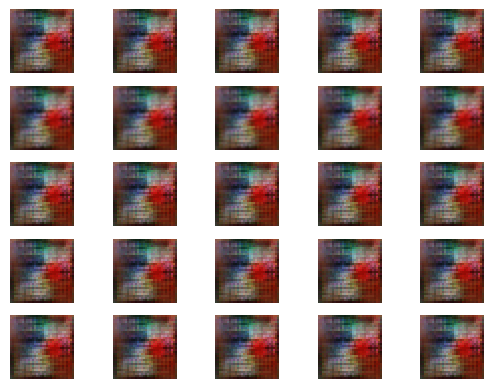
epoch\_3000 epoch\_4000 epoch\_5000



epoch\_6000 epoch\_7000 epoch\_8000



epoch\_9000



4)

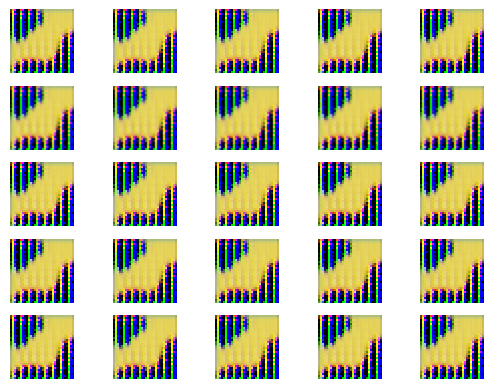
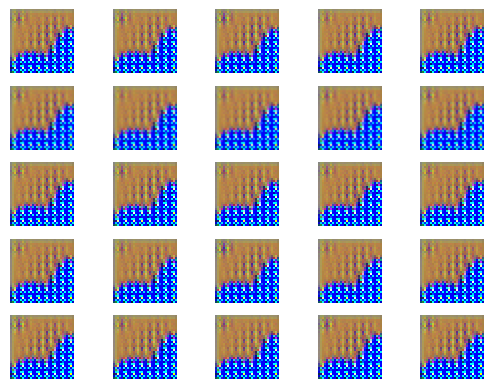
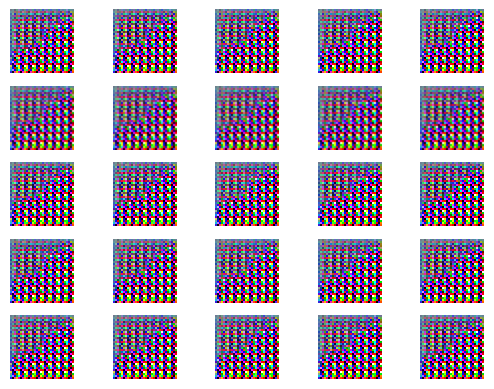
**Generator**,

* **Added Layer**: One Conv2DTranspose layer with **32 filters**.

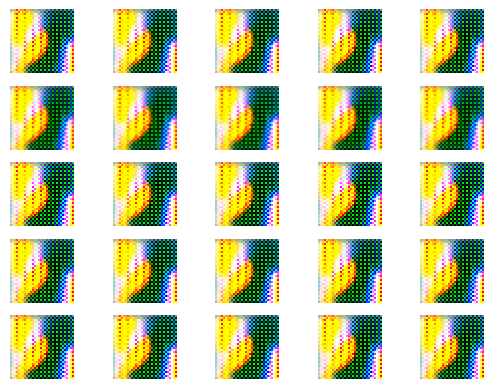
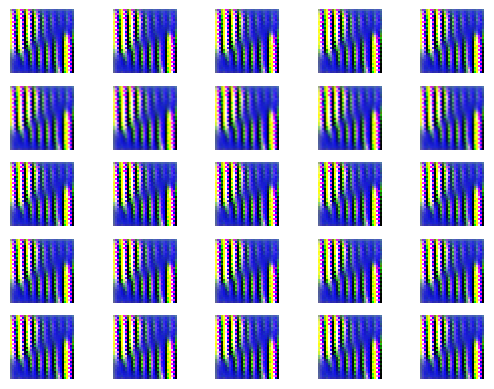
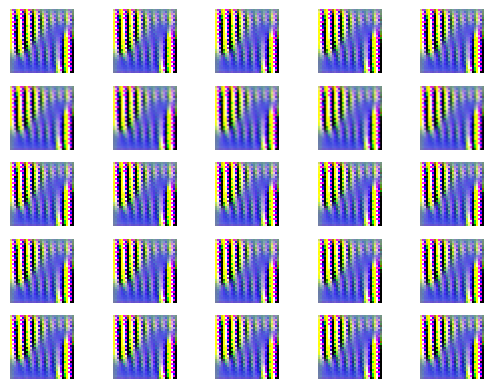
**Discriminator**,

* **Added Layers**: Two Conv2D layers with **256 filters** and **512 filters**, respectively.

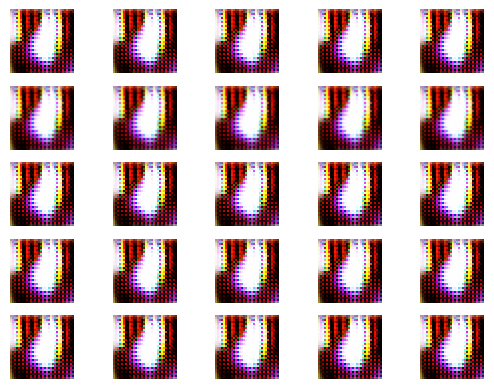
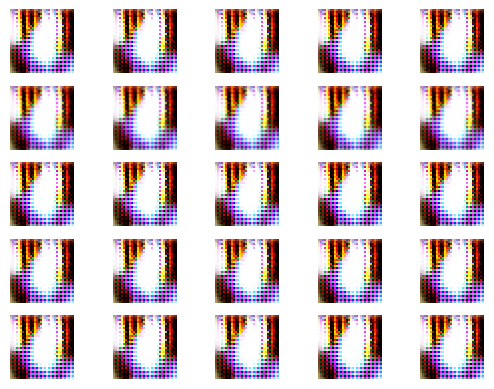
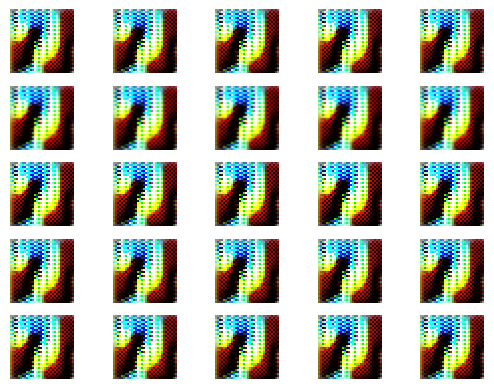
epoch\_0 epoch\_100 epoch\_200



epoch\_300 epoch\_400 epoch\_500



epoch\_600 epoch\_700 epoch\_800



epoch\_900

