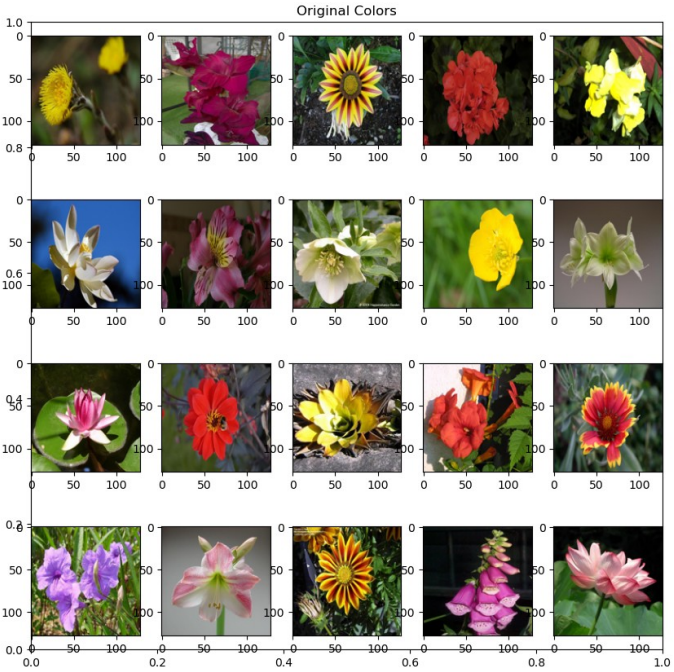
Deep Learning Final Project Report

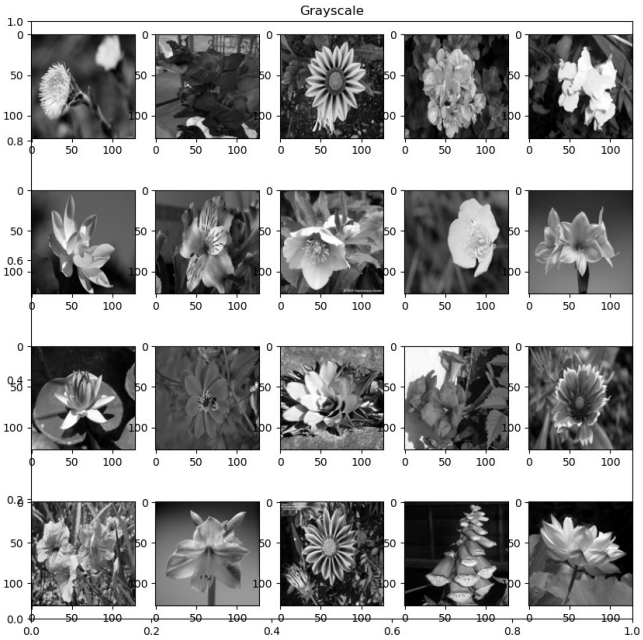
Barr Israel – 321620049
Moanes Samara – 315284349

Model Download:
https://drive.google.com/file/d/1KehOqy17oNfODTW5zNkr2fYGcRwfwtnE

Goal:
Our model aims to color grayscale images in a realistic way.
We focused on coloring images of flowers.

Dataset: 8188 different colored images of flowers from the following dataset:
https://www.robots.ox.ac.uk/~vgg/data/flowers/102/
a randomized(seeded, random and not basic split because the images are generally grouped by flower type) 2% of the dataset was used as a test set, the rest was used for training.
We chose 20 images at random(seeded) from the test set for our examples:

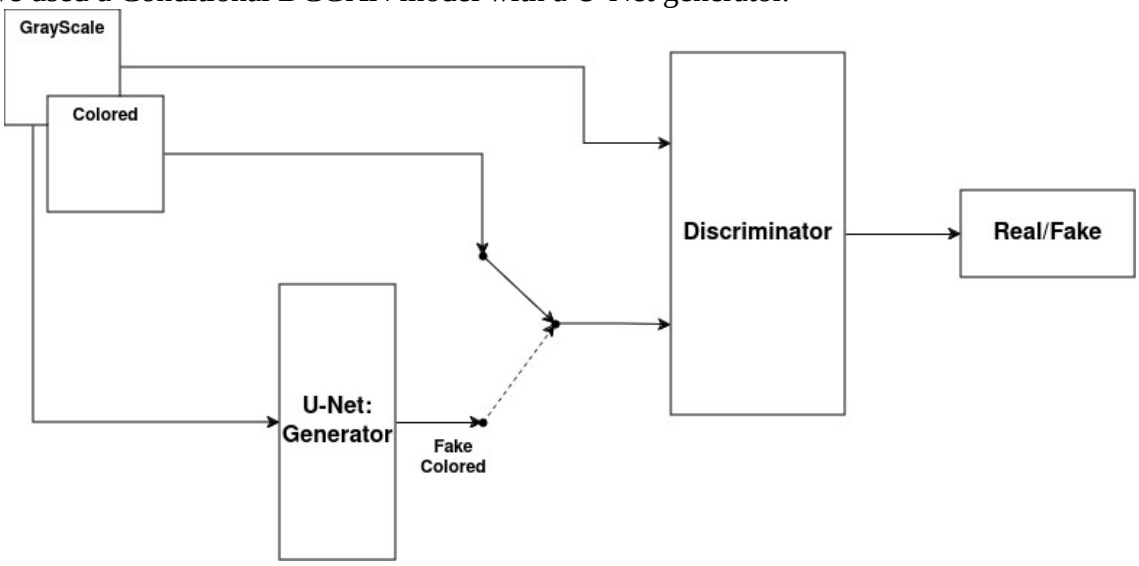




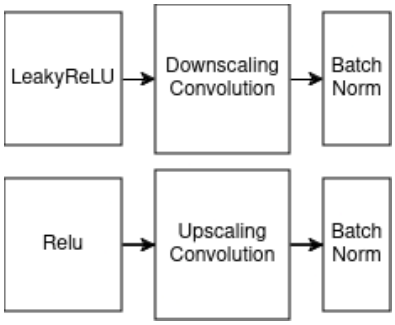
Colorspaces:
we tested training the model on multiple colorspaces: RGB, HSV and YcbCr.
The benefit of using the non-rgb colorspaces is that grayscale is already one of their 3 values so we only need to predict 2 values per pixel.

Preprocessing:
Each image was resized to 128x128 and normalized from 0-255 uint8 to 0-1 float16, in each of the 3 tested colorspaces.

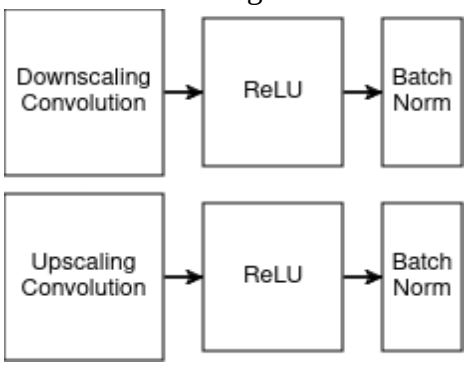
Architecture:
We used a Conditional DCGAN model with a U-Net generator.



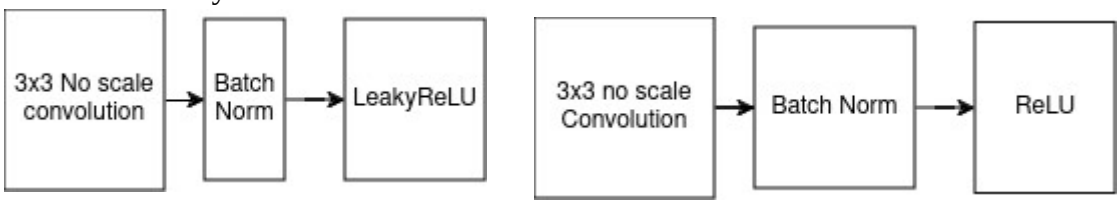
Our model is inspired by the models in “Colorful Image Colorization”[1] and “Image-to-Image Translation with Conditional Adversarial Networks”[2].
“Image-to-Image Translation with Conditional Adversarial Networks” suggests the following convolution blocks:



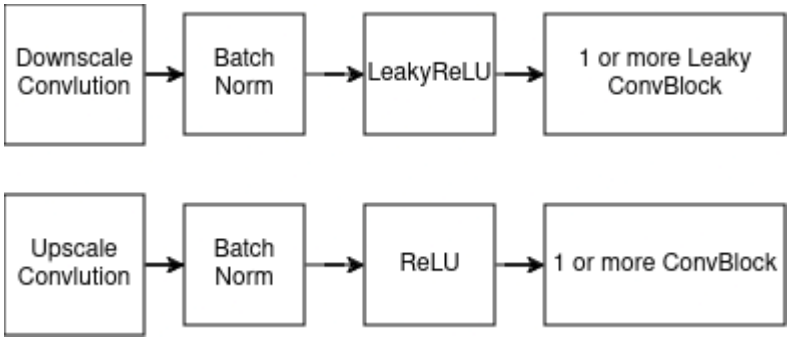
And Colorful Image Colorization suggests the following convolution blocks:



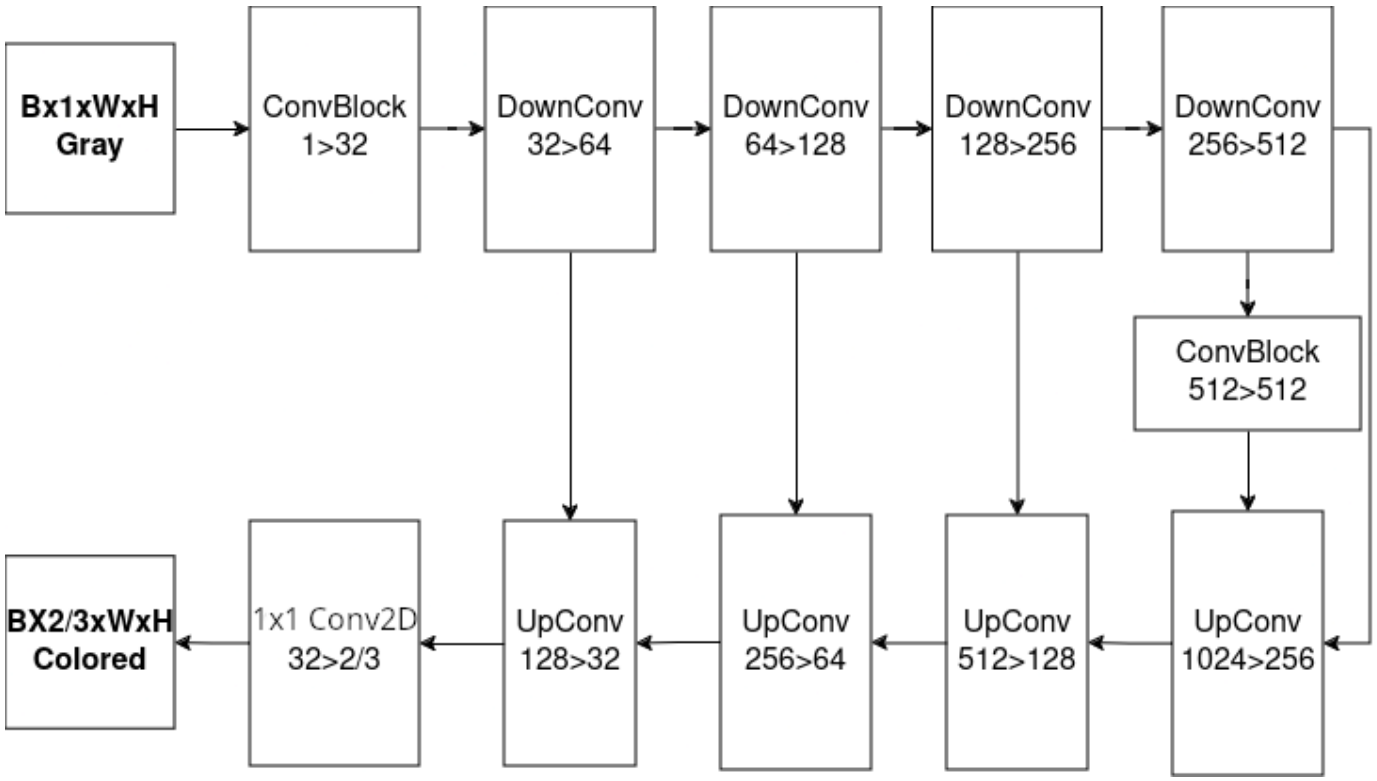
Our models introduced the following blocks:
ConvBlock/ Leaky ConvBlock:



Scaling Blocks:



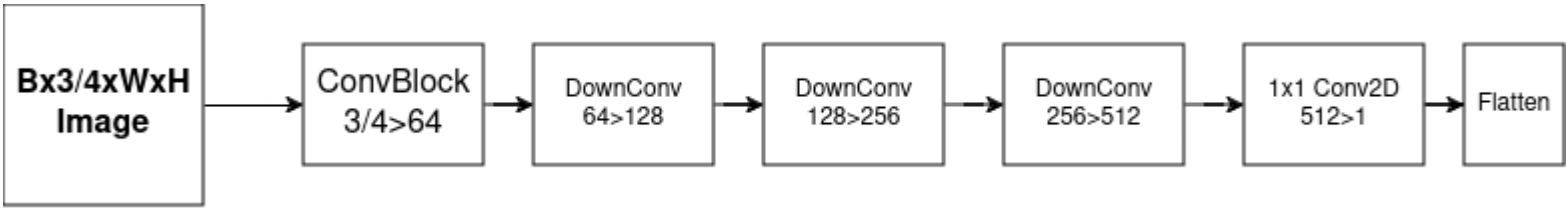
The full U-Net architecture for the generator is as follows:



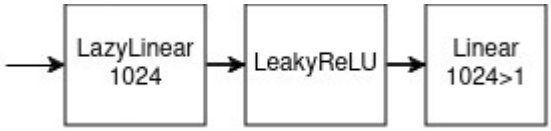
For the discriminator we evaluated 3 different architecture:

- Conv+Linear: Convolutional network+Dense Layers for final single real/fake scalar:
- Patch: Convlutional network for per patch scalar(as used in Image-to-Image Translation with Conditional Adversarial Networks[2])
- U-Net: U-Net discriminator

Both the Conv+Linear and the Patch discriminators used the same convolutional layers:



The Conv+Linear added the following dense layers:



The U-Net discriminator used the same architecture as the generator but accepting input in 3/4 dimensions(depending on the colorspace, gray+RGB/gray+the other 2 dimensions in HSV/YcbCr) and outputting a single layered real/fake output, without the final sigmoid layer(added as neccesary with a fused BinaryCrossEntropy loss function)

Learning: Both models used the Adam optimizer.

the generator loss function was a sum of the discriminator loss and a weighted L1 loss from the original colored image.

For the discriminator we evaluated both Cross Entropy loss(with label smoothing) and Wasserstein loss.

All models were trained on a laptop using an RTX 2070 Max-Q with 8GB of VRAM.

Evaluation:

The main evaluation metric we chose is FID(Fréchet Inception Distance) using the torcheval implementaion.

FID measures both the similarity of the generator output to the expected output, and the variety of the images in the batch.

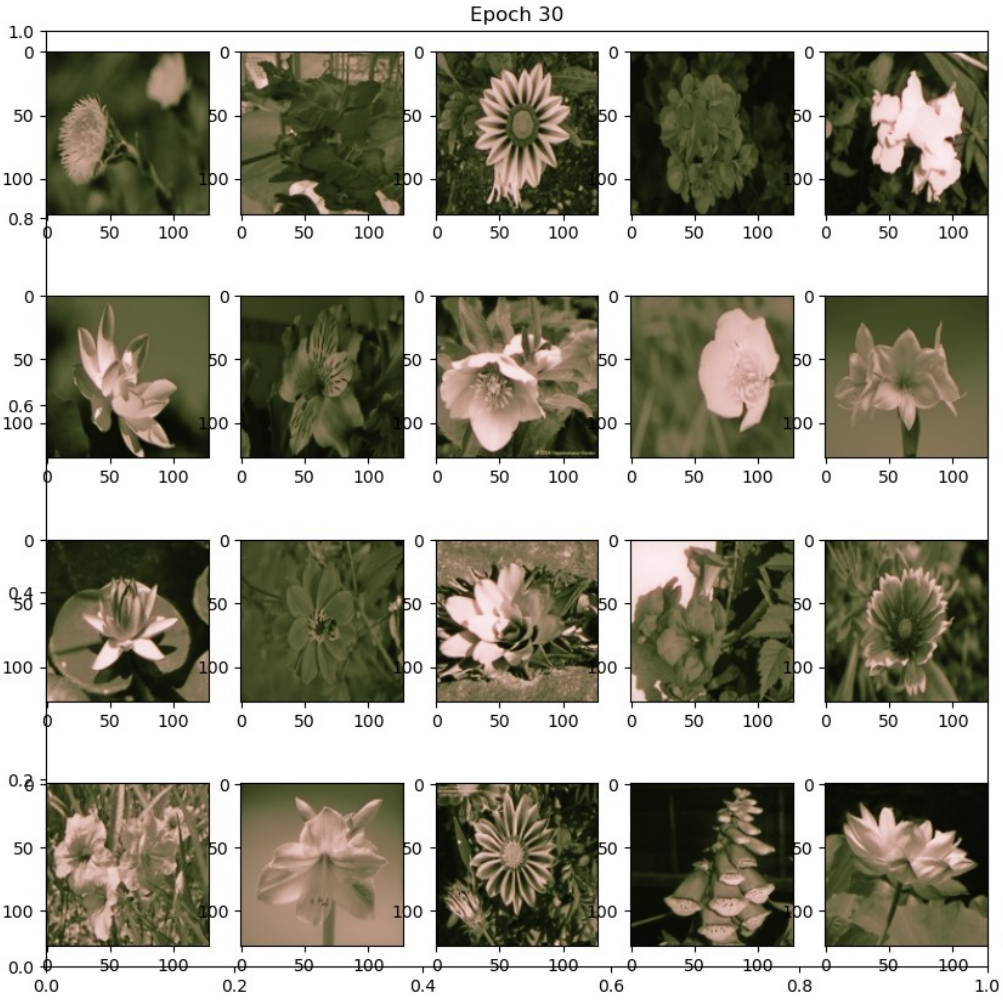
A low variety indicates the generator learned a specific coloring that fools the discriminator instead of learning the true way to color the images.

Because FID is expensive to calculate, it was only used on the test dataset, and L1 was used on the training steps.

<u>Choosing The Best Model:</u>

Simple U-Net Without GAN:

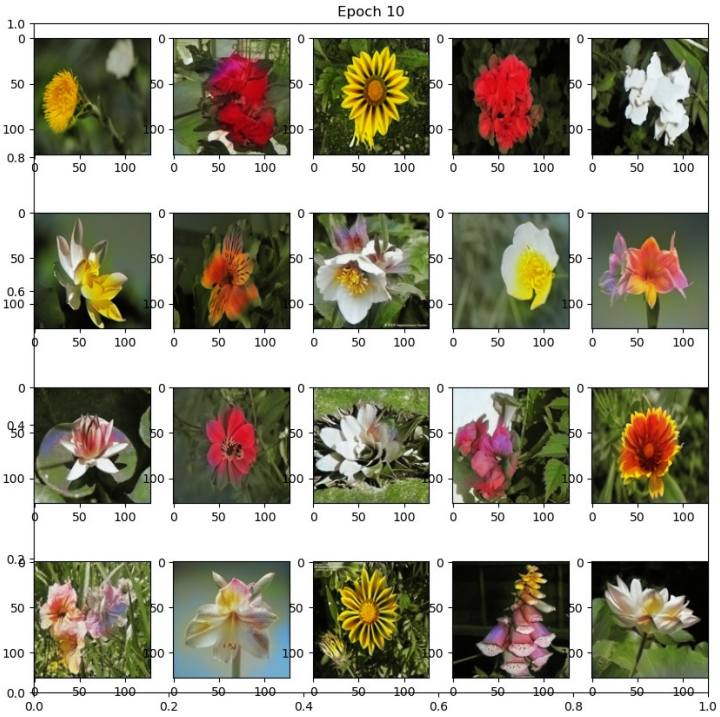
We first tried training a model without GAN, it was trained for 30 epochs(~20 minutes)
Final FID=87.99
The model fails to capture colors and mainly tints the original input, the results are very bad.



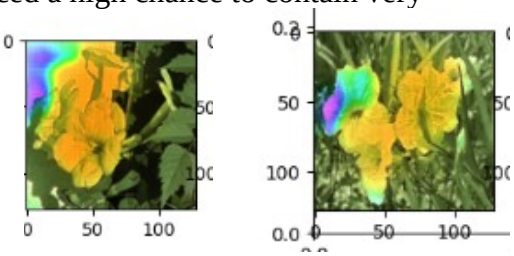
GAN Models:
The following GAN models were trained using L1 Weight=100, batch size=64 and for 10 epochs.
The Conv+Linear and Patch models had ~20M parameters and trained for ~15 minutes.
The U-Net discriminator had ~30M parameters and trained for ~16 minutes(the big increase in parameter count did not significantly increase the training time).
Conv+Linear RGB: Final FID=68.78, Visually very clear boundry problems

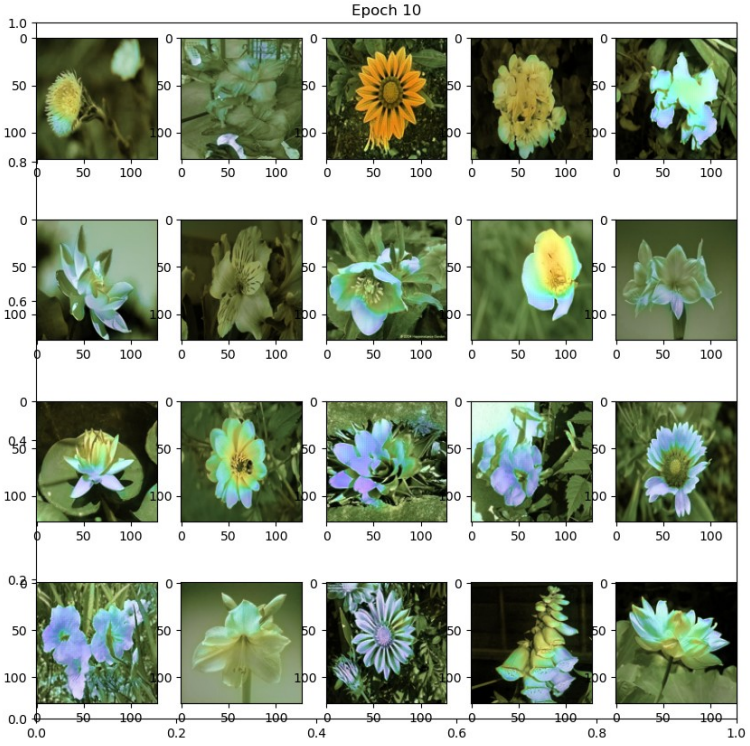
Patch RGB: Final FID=55.38, Visually significantly better than Conv+Linear RGB

U-Net discriminator RGB: Final FID=59.8, visually slightly better in some pictures and slightly worse in others compared to Patch RGB, specifically handles boundries better than any other model.

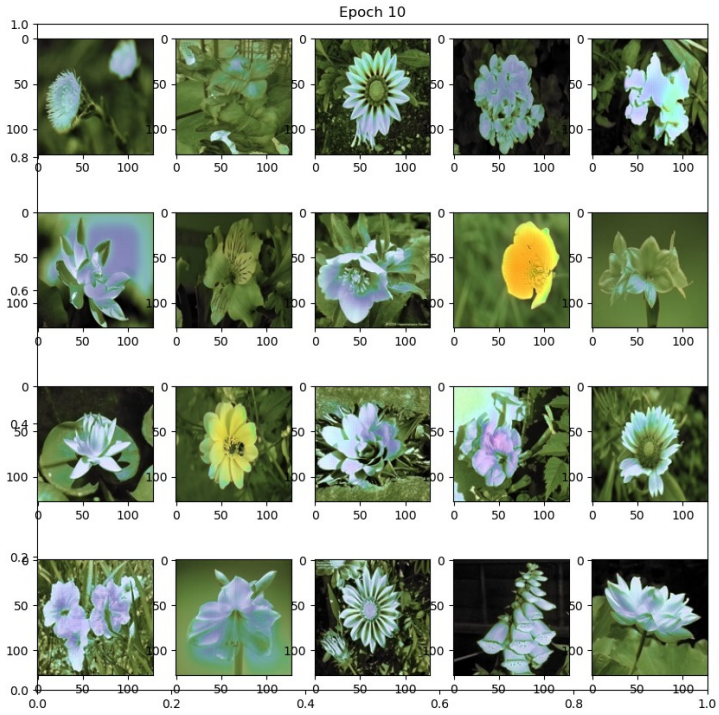


Conv+Linear HSV: Final FID=64.45, visually does not have the same boundry issues as Conv+Linear RGB but bad colors and texture, during training noticed a high chance to contain very badly colored areas in all HSV models:

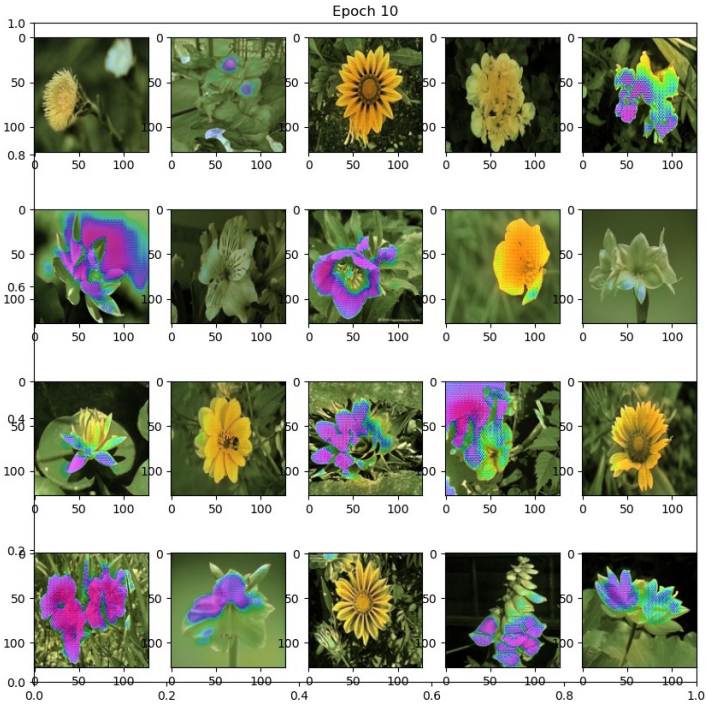


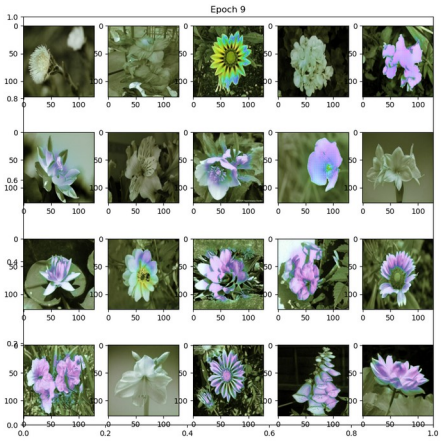


Patch HSV: Final FID=68.42, visually slightly better than Conv+Linear HSV but much worse than Patch RGB

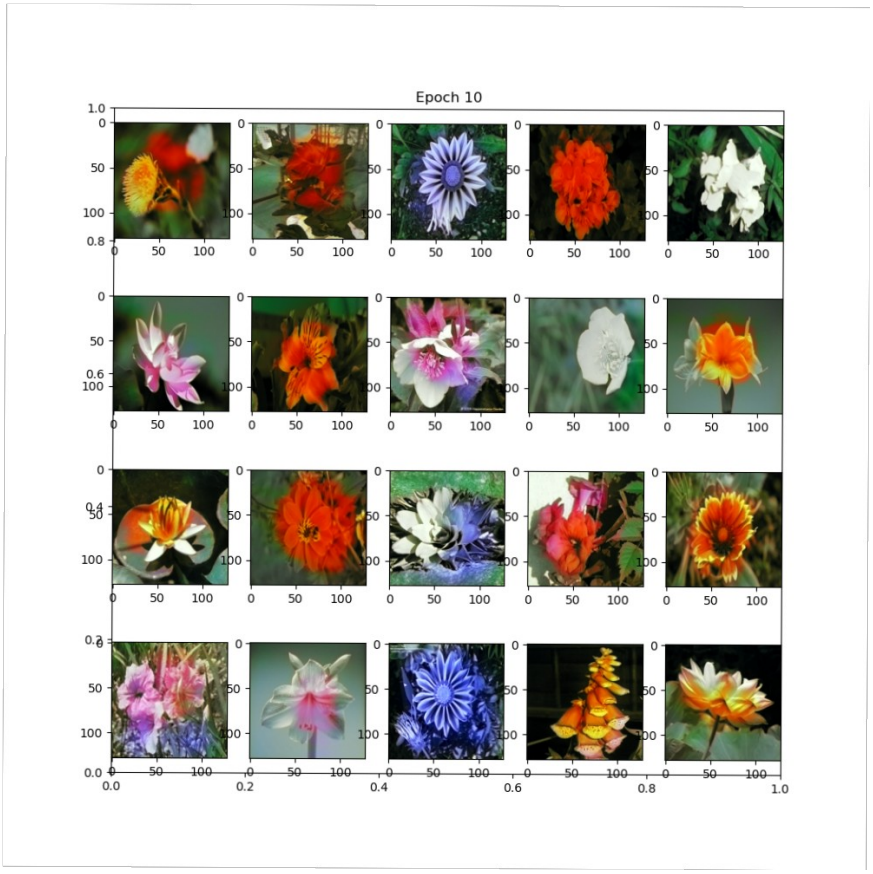


U-Net discriminator HSV: Final FID=92.19, Visually very bad final results, epoch 9 and a few others showed better results, but still bad:

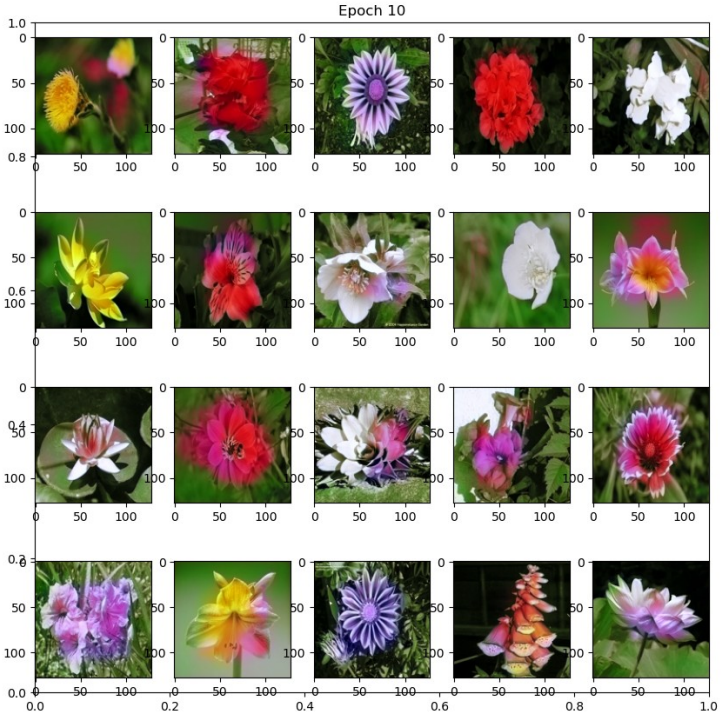




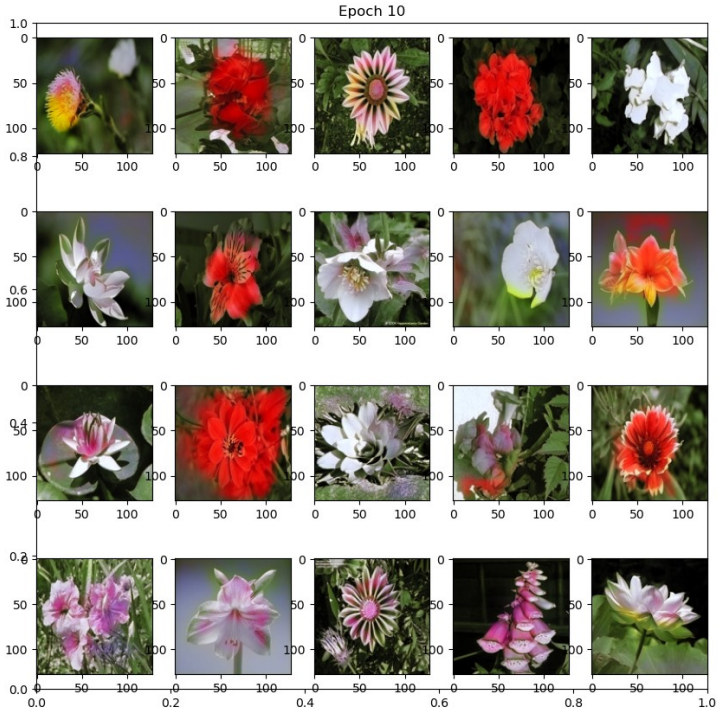
Conv+Linear YCbCr: Final FID=62.7, Visually about the same as Conv+Linear RGB



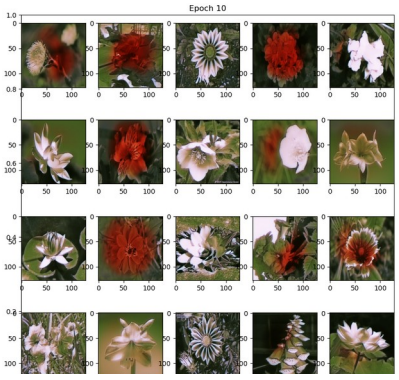
Patch YCbCr: Final FID=59.67, Visually better than Conv+Linear YCbCr and slightly better than Patch RGB

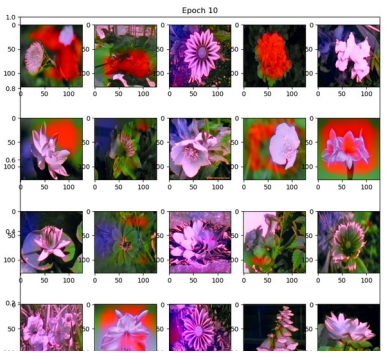


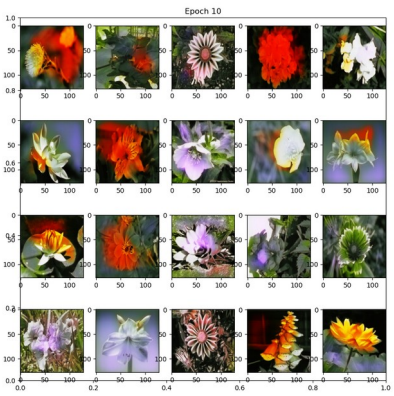
U-Net discriminator YCbCr: Final FID=54.73, the best FID despite not being the best visually, visually slightly worse than Patch YCbCr, worse than U-Net RGB

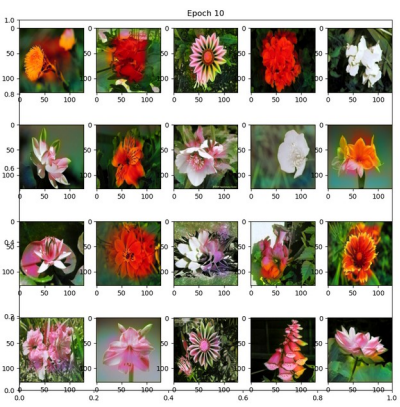


Based on these results, we stopped testing HSV and Conv+Linear models because they perform strictly worse than the rest.
Next we tested the usage of Wasserstein loss for the discriminator on the remaining 4 models but all of the models got worse
PatchRGB:71.78,UnetRGB:75.19,PatchYCbCr: 66.23,UNetYCbCr: 52.77 The UnetYCbCR FID score is appears to be an anomaly compared to previous epochs and it is also visually bad.





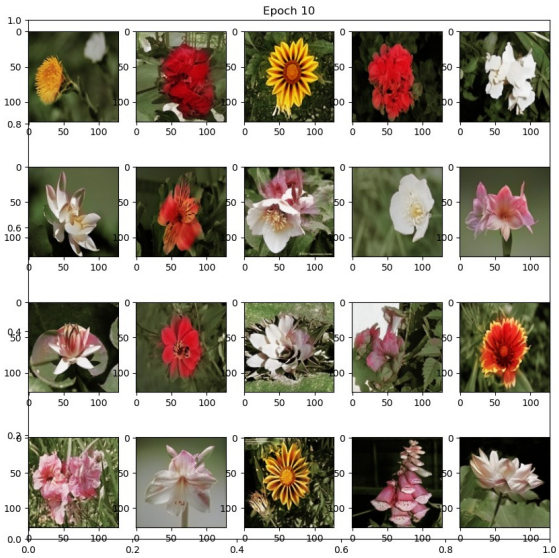




The final models chosen was the U-Net RGB despite not achiveing the lowest FID, based visually on its ability to handle boundries.

Hyper Parameter Optimization:

After testing a range of values for the L1 weight(1-1024), we chose 512 based on its FID=54.2 and its visual appearance:



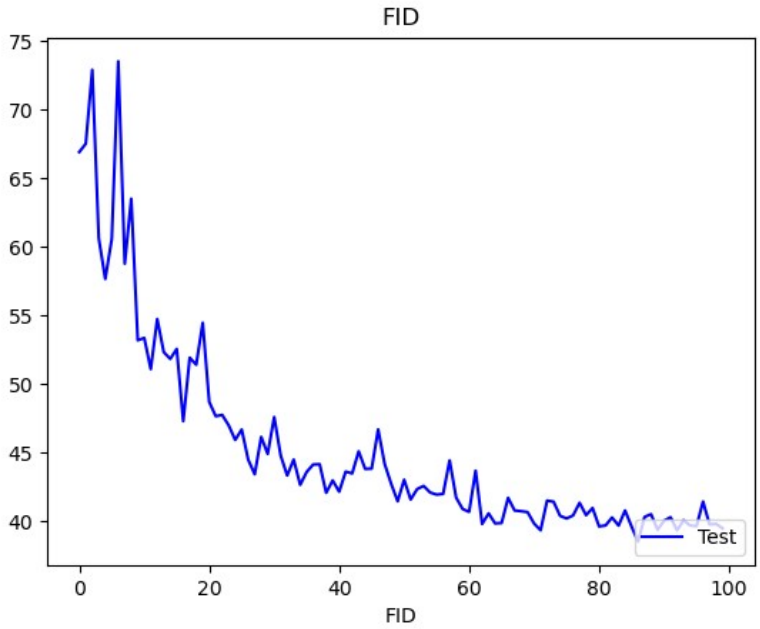
Other failed optimization attempts:

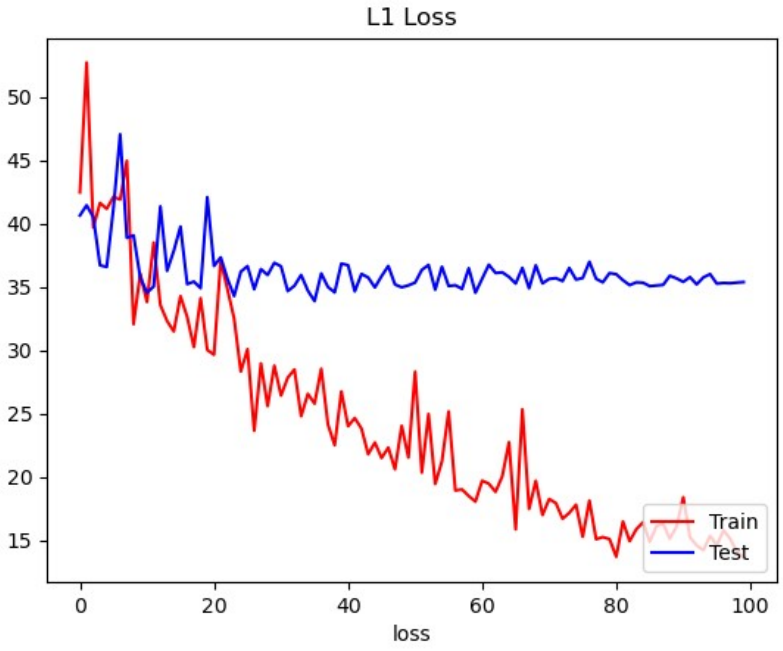
- increasing the amount of ConvBlocks in each Down/Up scaling did not show a noticable improvement.
- Truncating the U-Net discriminator made the boundries very slightly better and the colors worse(the idea was a middle ground between U-Net and Patch)
- Widening each layer of the discriminator did not show an improvement but expectedly increased the training time a lot.

Final Results:

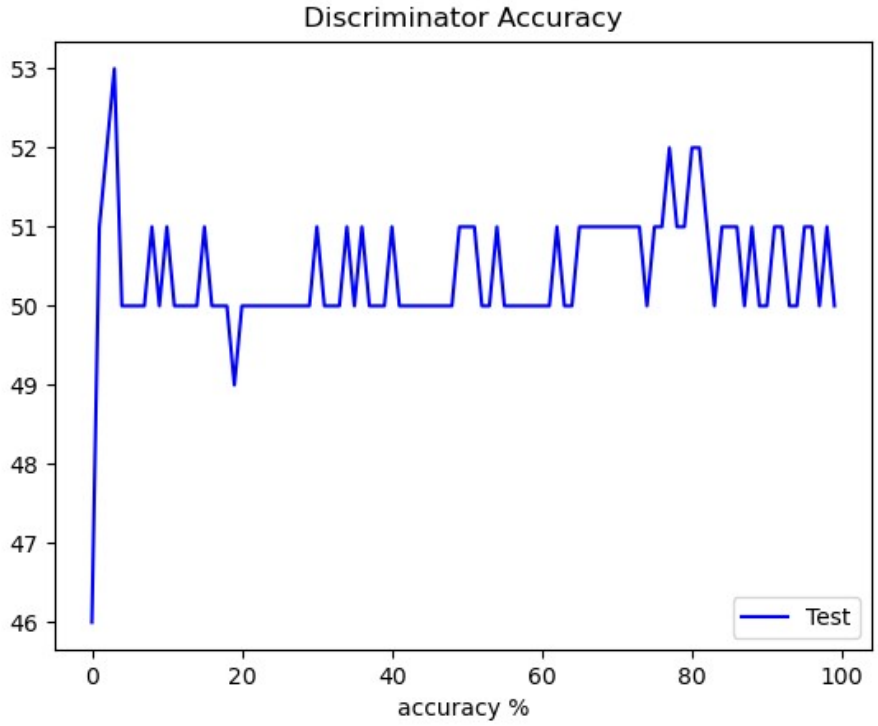
To reiterate, the final model chosen was the U-Net RGB model with L1 Weight=512 and no further changes.
The final model was trained for 100 epochs(~2:40 hours) and it has 29,279,268 parameters.
The model achieved a final FID of 39.51 and appeared to have reached or nearly reached its saturation point.

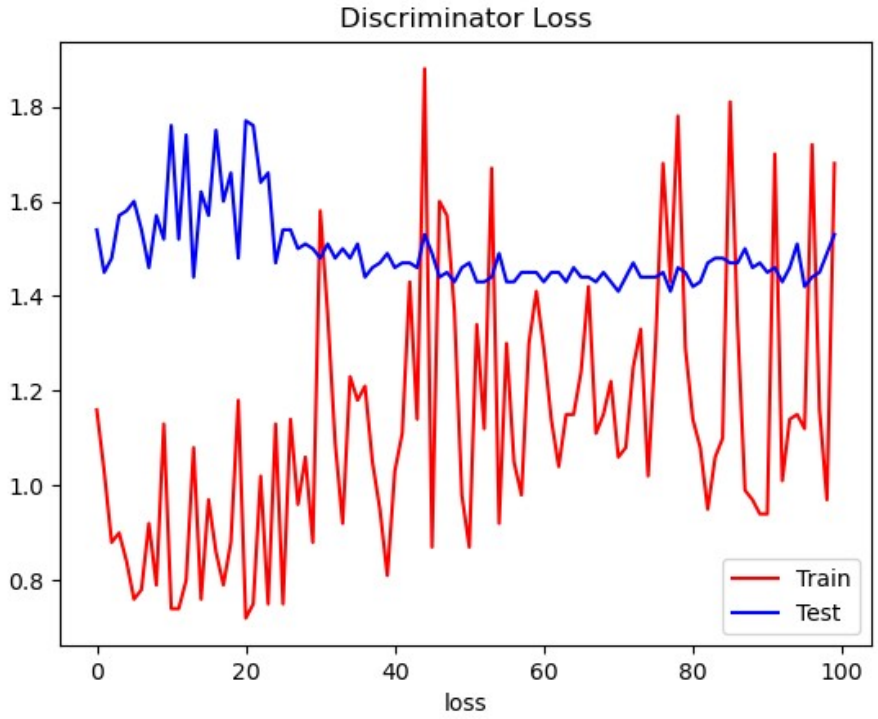
Based on the FID graph, after ~60 epochs the model can hardly improve further, Reminder: FID was only an evaluation metric and did not participate in any loss function.
Despite the L1 test loss flattening after ~20 epochs, the FID kept improving, which suggests the coloring kept improving, and suggests that the simple L1 loss with no GAN used at first really is not not sufficient.



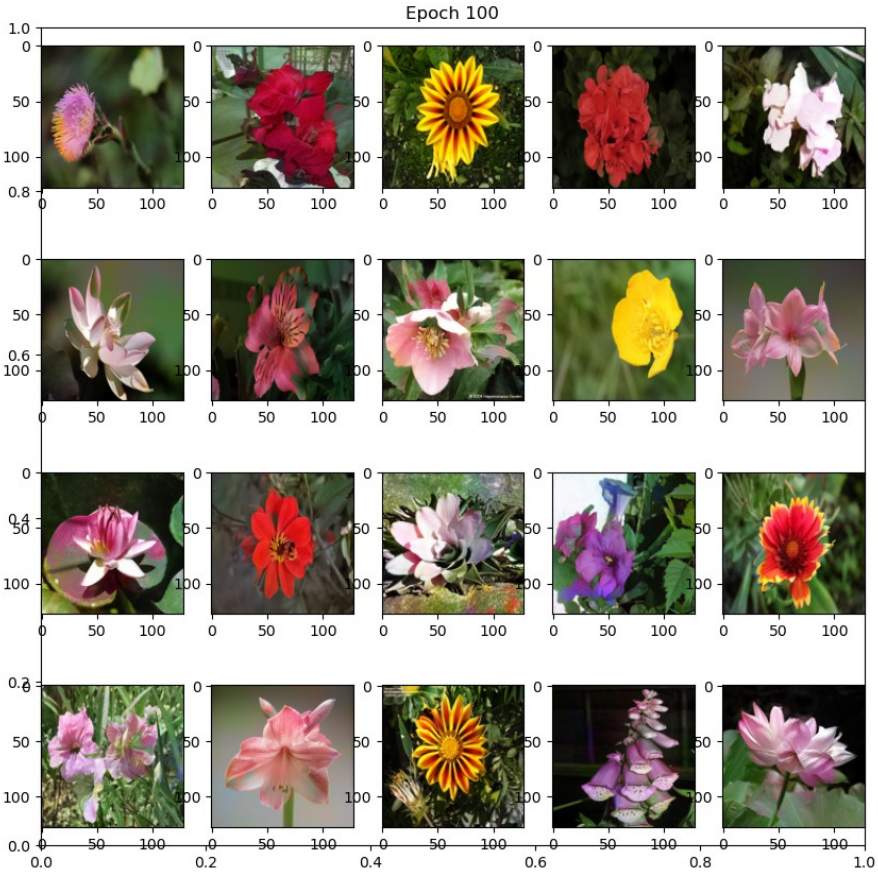


The discriminator results show that the generator fools it succesfully, which suggests that the discriminator could be improved to improve the overall results.

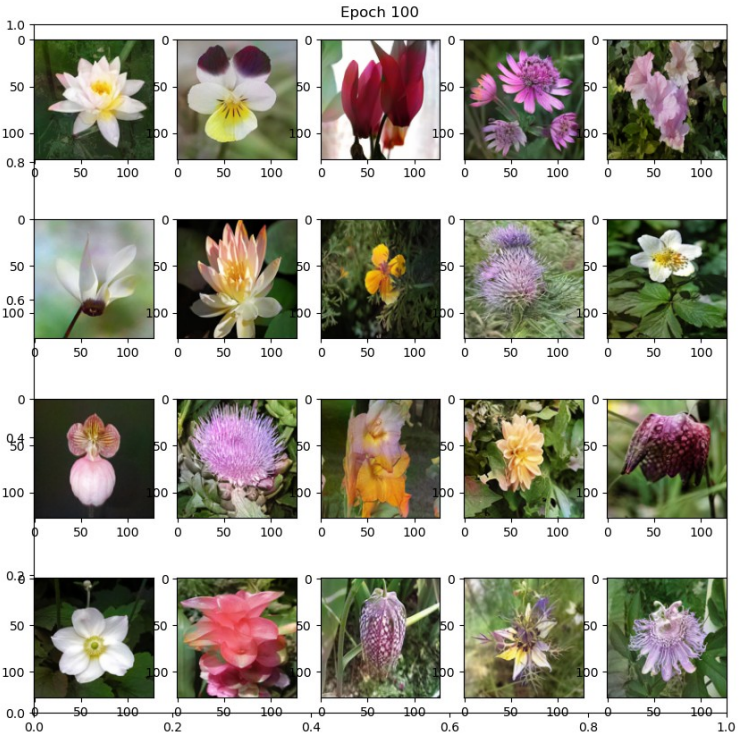




The output from the model using the original sample set:



The output from the next 20 images in the test set:



Further Possible Imrpvements:

- Based on “Influence of Color Spaces for Deep Learning Image Colorization”[3], a loss function for alternative colorspaces that first converts the image back to RGB can improve the results using these alternative colorspaces.
- Improving the discriminator network further as it appears to be a bottleneck based on its ~50% success rate, despite massively improving the original results.

Summary:

- The model we designed and trained achieves a satisfiable and realistic looking colored images in almost all examples.
- As the model appears to have reached it saturation point, more training without changes to the model or a different dataset is unlikely to noticably improve the results.

References:

1. Colorful Image Colorization, authors:Zhang, Richard and Isola, Phillip and Efros, Alexei A(2016)
2. Image-to-Image Translation with Conditional Adversarial Networks, authors:Isola, Phillip and Zhu, Jun-Yan and Zhou, Tinghui and Efros, Alexei A(2017)
3. Influence of Color Spaces for Deep Learning Image Colorization, authors:Coloma Ballester, Aurélie Bugeau, Hernan Carrillo, Michaël Clément, Rémi Giraud, Lara Raad, Patricia Vitoria(2022)