

## CENG 796 - Peer-review form

**Reviewed project ID:** Group #08

**Reviewed project's title (title of the paper):** HistoGAN: Controlling Colors of GAN-Generated and Real Images via Color Histograms

**Reviewer name(s):** Abdüllatif AĞCA, Ahmet Taha ALBAYRAK

Instructions:

- Answer = *Yes*, *No* or *Partial*.
- You may expand sections as necessary.
- For most questions, you do not need to add comments, unless the instructions tell you otherwise.
- "Notebook" refers to "Jupyter Notebook" file that is expected to be named as main.ipynb

Question	Answer	Comments
Contains a jupyter notebook file	Yes	
Notebook is located at <project_root>/main.ipynb	Yes	
Notebook's first section contains paper information (paper title, paper authors, and project group members' name & contact information)  Some good examples: see group03, group10, group11 (and a couple of other groups).	Yes	
Notebook contains a section for hyper-parameters of the model.	Yes	They mentioned activation functions, number of layers, learning rate, # epochs, batch size etc. in the code section as well as in some of the figures they attached.
Notebook contains a section for training & saving the model.	Yes	They saved the generator and discriminator separately as "generator.pt" and "discriminator.pt"
Notebook contains a section (or a few sections) for loading a pre-trained model & computing qualitative samples/outputs.	Yes	The code tries to find the latest checkpoints first and if succeeds, continues training with these parameters. Starts from scratch if fails to do so.
Notebook contains reproduced plots and/or tables, as declared.	No	There are failed (saturated) generated outputs under "Saturation of Generator" title and also discriminator and generator logs as command line output given training percentage.
Notebook contains pre-computed outputs.	Yes	

Data is included and/or a proper download script is provided.	Yes	We couldn't find any automatic download script <b>but the download link to the related dataset has been provided in the main.ipynb file.</b> The link is as follows: <a href="https://www.kaggle.com/datasets/splcher/animefacedataset">https://www.kaggle.com/datasets/splcher/animefacedataset</a>
Notebook contains a section describing the difficulties encountered.	Yes	
The paper has achieved its goals and/or explained what is missing.	No	The experimental goals file could not be found. However, the results don't seem to be finalized, and the training doesn't seem to be going well (generator loss is a huge positive number while discriminator loss is a huge negative number, they didn't seem correct to me)
The notebook contains a section that reproduces the figure(s) and table(s) declared in the goals.	No	
The notebook also reports the original values of the targeted quantitative results, for comparison.	No	
MIT License is included.	Yes	
As the reviewer(s), you have read the paper & understood it.	Yes	
Implementation of the model seems correct.		<p><b>The residual block</b> implementation seems to be correct (the figure in the paper was referred by the programmer as a comment, Figure S1)</p> <p><b>Discriminator</b> network was implemented as Supplementary Material, section 5.1 mentions (7 res blocks, which is <math>\log_2(256) - 1</math> as mentioned in the paper + a dense layer). Output channels for each res block is also correct</p> <p><b>ModDemodConv3x3</b> also seems to be corrected</p> <p><b>ModConv1x1</b> is correct</p> <p><b>HistoGAN:</b> From the code, I could not find the encoder and decoder blocks, constructing the body of the HistoGAN. Also I could not see the HistoGAN blocks constructing the "head" part of the HistoGAN. These blocks are demonstrated in Figure S2 and the HistoGAN's architecture is shown in Figure 4. <b>In their challenges section, they mentioned that the HistoGAN did not produce satisfactory results, so they switched back to StyleGAN</b> ("We saw that generator of HistoGAN does not learn and tried to train StyleGAN (with the shallow version that HistoGAN uses, it produces 256x256 images). However, we get rapid saturation of generator and have not solved yet. Here we present some generated images from training..."), <b>so we're not sure if it's related to it.</b> Also, HistoGAN's generator seems to accept image and output another image with the target histogram. However, the training code seems to get z (torch.randn, which is a normalized vector) and target histogram as inputs, which doesn't seem like an image to image conversion.</p>

Notebook looks professional (in terms of notation, readability, etc.)	Yes	
Source code looks professional (in terms of coding style, comments, etc.)	Yes	

**Additional comments:**

- The code was written in a clean way, and it was self-explanatory, but the comments made it even better by referring to the paper all the time, letting us find the figure/formula to compare the code easily.
- Training loop can be written in “.py” file so that we can see less code during analyzing the outputs. It can be a good refactoring practice in terms of readability.