CENG 796 - Peer-review form

Reviewed project ID: Group 02

Reviewed project's title (title of the paper): Drop the GAN: In Defense of Patches Nearest Neighbors as Single Image Generative Models

Reviewer name(s): Yusuf SOYDAN, Bartu AKYÜREK

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	
Notebook contains a section for training & saving the model.	No	The model GPNN defined in the notebook is implemented with a model that does not require any training process because the model has a patch nearest neighbor algorithm which is a variation of K-NN. Therefore, there is no saving for the model either.
Notebook contains a section (or a few sections) for loading a pre-trained model & computing qualitative samples/outputs.	Partial	Because the study does not contain any parametric model, the group members do not use any pre-trained model in the notebook. They show their result of SIFID score calculation with the original one stated in the paper. Due to the memory requirements they mentioned, they get a coarser FID score.
Notebook contains reproduced plots and/or tables, as declared.	Yes	

Notebook contains pre-computed outputs.	Yes	
Data is included and/or a proper download script is provided.	Partial	According to the notebook, the input dataset is moved to /real directory. 50-image input set is included in the repo
Notebook contains a section describing the difficulties encountered.	Yes	The group members declare the difficulties that they have encountered such as insufficient memory they run on and no clear definition is available in the paper for the model hyperparameters.
The paper has achieved its goals and/or explained what is missing.	Yes	The group members explain the mean calculation to be included for version-2.
The notebook contains a section that reproduces the figure(s) and table(s) declared in the goals.	Yes	
The notebook also reports the original values of the targeted quantitative results, for comparison.	Yes	
MIT License is included.	Yes	
As the reviewer(s), you have read the paper & understood it.	Yes	
Implementation of the model seems correct.	Partial	1. They compute single image FID (SIFID) score of the real and fake images by using sifid_score code downloaded from https://github.com/tamarott/SinGAN. 2. In aggregate_patches method, the paper suggests to use gaussian weighted-mean method, but in the code arithmetic mean is used (mentioned in notebook comments as well). 3. `compute_distance_between_images_pytorch` method is available but unused, due to GPU memory limits. 4. No need to calculate separate Q and V for n=0 in gpnn method, deepcopy or another method can be used for code clarity. 5. No need to use image argument in aggregate_patches method, rather image.size can be enough to save memory (image is tensor, image.size is tuple). 6. The only equation in the paper, eq.1, is implemented at the end of `compute_distance_between_images`. 7. The steps of Figure 3 can be followed throughout the notebook: • Patches are extracted at `gpnn()` • Distances are computed at `pnn()` • Normalized scores are computed inside of `compute_distance_between_images()` • Nearest neighbours are found at second line of `pnn()` • Patches are combined at the end of `pnn()`

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 paper proposes an alternative non-parametric k-NN like method to generate images by some kind of super-resolution technique. In the proposed method, the original image is downsampled gradually down to a certain size. At the last size, gaussian noise is injected into the downsampled image. The method benefits from this process for image generation with increasing sizes. The image generation is done by finding the cross MSE of query (Q) and key (K) patches; filtering the minimum MSE indexed patches from value (V) patches; calculating the pixel value generated as the output by weighted Gaussian sum of square distance (SSD) method between filtered V and downsampled original image for that level. The generation submethod is done for several iterations in order to incorporate the variations for the image generated. V is downsampled of the original image; K is the upsampled version of the downsampled original image for a certain step of the pyramid; Q is the generated image patch nearest neighbor method described above.

The image generation is operated by noise injecting a low-resolution downsampled of an image and applying this coarse image to a variation of super-resolution technique to reach higher resolutions. We consider that the method uses noise injection at the coarsest level of the downsampled image for the sake of simplicity. The authors of the paper could purpose that the noise injection at low resolution had to enhance the higher resolutions of the generated image by iteratively generating the same size image by using Gaussian-weighted SSD. However, the group members do not write Gaussian weighted SSD, instead, the code has arithmetic pixel mean calculations in the aggregation method of the generation process. Therefore, we consider that the images with lower intensities could possibly be generated with the code given in the notebook, and suggest to the group-02 members to attempt to write Gaussian SSD to generate higher intensity images which also concludes with a finer FID score.