CENG 796 - Peer-review form

Reviewed project ID: Group **4**

Reviewed project's title (title of the paper): SeD Semantic-Aware Discriminator for Image Super-Resolution

Reviewer name(s): Melih Gökay Yiğit - Umut Özyurt

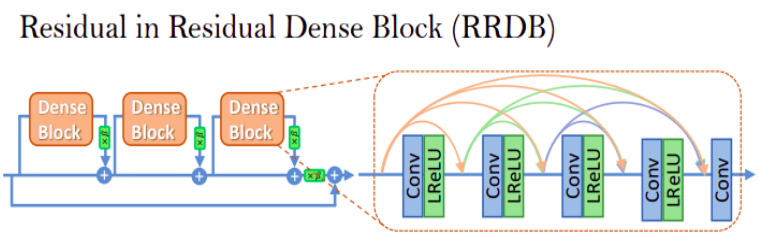
Instructions:

- 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).	Partial	• Paper author information is not provided.
Notebook contains a section for hyper-parameters of the model.	Yes	
Notebook contains a section for training & saving the model.	Yes	
Notebook contains a section (or a few sections) for loading a pre-trained model & computing qualitative samples/outputs.	Partial	• We could not try the notebook due to the problems on downloading datasets (big size and low download speed), one restricted download link and path issues (elaborated below).
Notebook contains reproduced plots and/or tables, as declared.	Partial	• Because of the technical issues they mentioned the instructor about, the declared plots are for a small subset of the dataset.

Notebook contains pre-computed outputs.	Yes	
Data is included and/or a proper download script is provided.	Partial	• Download script is problematic for the reasons stated in the additional comments part below.
Notebook contains a section describing the difficulties encountered.	Partial	• Why single head cross attention is used instead of multi head cross attention? (Why num_heads is 1?) • It is not written in the assumptions, but the feature maps are assumed to have a dimension of 128, which we cannot see in the paper explicitly. Is there a reason for this? (Why “dim_head” is 128?) • Is the choice of lambda values in the loss as 1 and 10 stems from a reason? For example, is it derived by hyperparameter experiments? We think this choice should affect the training significantly and the reason of lambda choices should have a reasoning.
The paper has achieved its goals and/or explained what is missing.	No	• Because of the technical problems they have encountered, they could not measure the metrics on the desired datasets.
The notebook contains a section that reproduces the figure(s) and table(s) declared in the goals.	Partial	
The notebook also reports the original values of the targeted quantitative results, for comparison.	No	• They are not explicitly stated in the notebook.

MIT License is included.	Partial	• 2 License files provided.
		  ○ License.txt is not MIT license.
		  ○ LICENSE contains only Yiğit Ekin’s name.

As the reviewer(s), you have read the paper & understood it.	Yes	
Implementation of the model seems correct.	Partial	• For RRDB module, leaky relu (or any activation) is not used between conv layers, which is in contrary to the referred RRDB paper. (https://arxiv.org/pdf/1809.00219) Related Fig4 figure is below:  In ”models/rrdb.py”, in ”DenseBlock” class, the LReLU activations in the implementation is missing. (since ”get_layer” function is used with ”is_upsamle=False” as default, and in the resulting if

		statement, there is only a convolution layer, without any activation. • Other parts (e.g. SeFB and its input-output connections, getting PVM's parts before its layer 4...) seemed to align with the paper.
Notebook looks professional (in terms of notation, readability, etc.)	Partial	• Plots and examples demonstrated well. We think that adding train.py files content to main jupyter notebook is not necessary.
Source code looks professional (in terms of coding style, comments, etc.)	Partial	• The code is lack of comments, which makes it hard to follow the code. It should contain more comments, possibly referring where it is related to in which paper for easier interpretability. • Also, using too much (e.g. RRDBNet -> Residual_in_ResidualBlock -> DenseBlock -> make_blocks -> get_layer -> Conv2d) hierarchy deteriorates the code's readability in our opinion. It may be 1-2 levels more basic to enhance following up the code.

Additional comments:

- In the implementations “models/patchgan_discriminator.py” file, there is an absolute path import “sys.path.append('/scratch/users/hpc-yekin/hpc_run/SeD/models')”, which should be changed with a dynamic path like “sys.path.append('./models')”, since the former can only work with your local pc, and the letter may be more flexible.

- The environment.yml file does not install “clip”, but it must be imported in “models/feature_extractor_model.py”.

- download_dataset.sh did not work with your “resize_4.py”. This is because parser should have "--folder" instead of "folder" and "--save_path" instead of "save_path" (-- is missing).
  The “gdown https://drive.google.com/file/d/1henrktM4Cw9hJIJBDEObAzl-eCbpzNaJ/view?usp=drive_link” part does not work since the link requires access. Moreover, the download of “Flickr2K” were very slow (lower than 500kb/s) in our case. Again, it may be a problem on our side, but it is likely to be caused from the enabled download rate from the given link. If some other faster link or option is provided in the script, it would be much better.

- In the “prepare_dataset.py” script, the “os.mkdir("data2/dataset_cropped") if not os.path.exists("data2/dataset_cropped") else None” line does not work. You can replace it with os.makedirs("data2/dataset_cropped", exist_ok=True)”.
  Similarly, replacing the ”os.mkdir(save_folder) if not os.path.exists(save_folder) else None“ with ”os.makedirs(save_folder, exist_ok=True)“ should work.

- Still, the script “prepare_dataset.py” does not work since it assumes there is a “data2/hr“ and “data2/lr” directories exist already. (we did not understand the “data2” folder. Maybe, it is left from the code authors’ local environment)

- In the “models/super_resolution_module.py” there is a possible redundant model instance creation at line 34. It should be merged with an if-else block with the line 36. Creating a “PatchDiscriminatorWithSeD“ instance can be somehow costly and if “is_pixelwise_disc“ is False, it will be created and then deleted for no reason.

- Also, in “models/super_resolution_module.py”, the lines 132 and 133 sends low and high resolution batch parts to GPU which remains unused, because “get_loss_forward_dict” method uses batch object. This causes unnecessary usage of GPU memory, which may limit the highest possible training batch size to 50%.

- Also, for SeD patchwise discriminator, the adversarial loss of the generator converged to a very high value too early, whereas the discriminators adversarial loss converged to a low value. This may be a sign of mode collapse or unstable training (since generators loss always increased).
  Similarly, for patchwise discriminator, the generators adversarial loss converges to a value which does not show significant improvement (decrease), whereas the discriminators adversarial loss seems to converge to a low value too early, may be caused from the same issues.

- In the discriminators initial part, paper only states a “conv” in the architecture. However, using a 4 block downsampler with stride=2 may cause an unbalance between the discriminator and the generator. One of the reasons of discriminator learns too much and generator learn nearly nothing might be due to this (reported plots of training adversarial losses in the notebook). We are suggesting to pay attention to discriminator experiments.