CENG 796 - Peer-review form

Reviewed project ID: Group **1**
Reviewed project's title (title of the paper): **HeadGAN: One-shot Neural Head Synthesis and Editing**
Reviewer name(s): **ALI IBRAHIM OZKAN – SEMIH SOGANCI**

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**	The first section contains the paper title and group members. However, it does not contain contact information and paper authors.
Notebook contains a section for hyper-parameters of the model.	**YES**	Notebook contains **Training Hyperparameters** section and hyperparameters such as learning rate, optimizer betas, epoch size were described here.
Notebook contains a section for training & saving the model.	**YES**	Notebook contains **Training & Saving Model** section. Its implementation is in the training.py file. The model is trained and saved in the training_loop function in the training.py.
Notebook contains a section (or a few sections) for loading a pre-trained model & computing qualitative samples/outputs.	**YES**	Notebook contains **Load the saved model and compute qualitative results** section. The epoch number of the model related to the qualitative results can be given.
Notebook contains reproduced plots and/or tables, as declared.	**N/A**	N/A
Notebook contains pre-computed outputs.	**YES**	Notebook does not have a specific section for pre-computed outputs. 3d face extraction masks are computed by 3DDFA v2 architecture for reference images. Extraction masks of pre-computations can be seen in the second column of qualitative figures section.

Data is included and/or a proper download script is provided.	**PARTIAL**	The project consists of download_data.sh. However, in order to download 3DDFA V2 and the model, we need to use the links in the readme.md file. Although these links are google drive links, they weren't included in the script. On the other hand, the vox2 dataset was downloaded via download_data.sh script after getting the username and password. Downloading this dataset is taking too much time, maybe a small subset of the dataset can be separately linked to the drive just for quick testing and demo purposes.
Notebook contains a section describing the difficulties encountered.	**YES**	It is mentioned that the main difficulty stems from the sizes of the datasets. You can improve and evaluate your results quicker when training over a small subset of datasets and get intermediate outputs from your separate networks for debugging purposes.
The paper has achieved its goals and/or explained what is missing.	**PARTIAL**	The goals.txt doesn’t contain the Version 1 submission part. However, the v1 and v2 submission goals and results are in the Challenges part.
The notebook contains a section that reproduces the figure(s) and table(s) declared in the goals.	**PARTIAL**	The notebook does not contain a section related to specific targeted qualitative figures. If it is not necessary to get outputs of same figures as in the paper, there is a notebook section showing the qualitative figures from the network. On the other hand, the figures are not that mature for that training level. You can show intermediate outputs (e.g. after dense flow network) for debugging purposes or you can train subnetworks separately so that you can show that your networks are working. You can also use low resolution pictures to get close outputs between the driven and reference images by reducing the computational costs. On the other hand, the notebook contains a table comparing quantitative goals and achieved results. Also in here, the results are not close since the model were not be able to be fully trained due to large datasets.
The notebook also reports the original values of the targeted quantitative results, for comparison.	**YES**	The notebook contains targeted quantitative results (PSNR, FID) which are above the Challenges part.
MIT License is included.	**YES**	
As the reviewer(s), you have read the paper & understood it.	**YES**	The paper proposes a GAN architecture for face reenactment; - by adding facial expressions and pose from a driving image using a dense flow network. - by adding audio features to construct mouth movements using a rendering network Only 3d pose enrichment is implemented by our friends in this study without the audio augmentation.
Implementation of the model seems correct.	**PARTIAL**	* The SPADE layer (Figure 9a) implementation seems correct in spade.py. * The Adain layer implementation(adain.py) seems different than Figure9b. Instead of gamma and beta values, mean and std values are used in implementation.

		* The Discriminator implementation (discriminator.py) is consistent with the referenced paper. (https://arxiv.org/pdf/1903.07291.pdf , Figure13) * Rendering network implementation(renderingnetwork.py) seems correct as in Table 5. However, it consists of duplicate variables (Line 53 to Line 59) * The Dense Flow Network scheme(Table 4) is compatible with the implementation (denseflownetwork.py)
Notebook looks professional (in terms of notation, readability, etc.)	**PARTIAL**	Notebook can be improved in terms of comments, paper explanations and model architectures. Also, input and output sizes of the computational blocks can be represented in a table both for the generator and the discriminator networks.
Source code looks professional (in terms of coding style, comments, etc.)	**PARTIAL**	Files were not grouped according to their operations. Only misc folder were created. The additional comments section was used for detailed comments. Mostly snake case convention is used in the project. The additional comments section was used for detailed comments. It is good to have testing sections at the end of each network file (e.g dense flow, discriminator etc..). You can also give comments about the shapes of the tensors in generator and discriminator models for the ease of traceability.

Additional comments:

Evaluation Part:

- Paper specifies the pytorch-fid implementation as a reference [35]. Using this implementation while calculating FID, may increase FID calculation accuracy.
    - The pre-trained Inception model which was used in the project may not capture all features. This can result in a high FID score

(https://wandb.ai/ayush-thakur/gan-evaluation/reports/How-to-Evaluate-GANs-using-Frechet-Inception-Distance-FID---Vmlldzo0MTAxOTI)

Loss Part:

- In Loss.py, the last term(line 78) **lambda_fm * L_F_VGG** loooks redundant. The loss function in paper doesn't contain this.

- Paper specifies uses the feature matching(F.M.) loss and reference paper stated as a [51]. This referenced implementation of F.M. loss is below, and it is different than implementation in HeadGan project.

Feature matching. To achieve more realistic results, we adopt a feature matching loss term [35], defined as

$$\frac{1}{N}\sum_{i=1}^{N}\|\phi_\theta^l(G_\omega(y^i)) - \phi_\theta^l(x^i)\|^2, \quad (5)$$

where $\phi_\theta^l(x)$ represents the feature response to input x at the l-th layer of the discriminator. This term forces the re-

- The adversarial loss stated in paper is below. The implementation is different than this loss function. The implementation uses BCE loss.

$$\mathcal{L}_D^{adv} = -\mathrm{E}_{p_{data}}[\min(0, -1 + D(\mathbf{x}_t, \mathbf{y}_t) - \min(0, -1 - D(\mathbf{x}_t, \tilde{\mathbf{y}}_t)].$$

The differences between loss in implementation and paper might affect the results.

General Recommendation

- After testing the files, the main function (if __name__ == "__main__":) of the files might be removed. The py files without main looks more professional.
- Some function has more than 3 parameters. With more parameters, the code might be harder to follow.
- Training.py consists commented codes and looks unprofessional.
- Files may be grouped into folders. For instance, the .py files about the dataset might be copied to the dataset.