NATIONAL UNIVERSITY OF SINGAPORE

EE5934 DEEP LEARNING PROJECT 2

Will you be the next Monet?

Deadline (Confirmed)
April 24th, 2022

Contact
songhua.liu@u.nus.edu

0 Academic Integrity

Academic integrity is the bottom line for all students and scholars engaged in academic activities. This course will spare no effort to maintain the academic integrity. Any violation of this bottom line will not be tolerated. Therefore, please **DO NOT PLAGIARIZE**! Softwares will be used to detect plagiarizing on both codes and reports and we will penalize it by giving zero score. You may refer to materials from Internet or some others, but do remember to make citations such that one can tell which part is actually yours. Submission of this project will be treated as agreeing this statement.

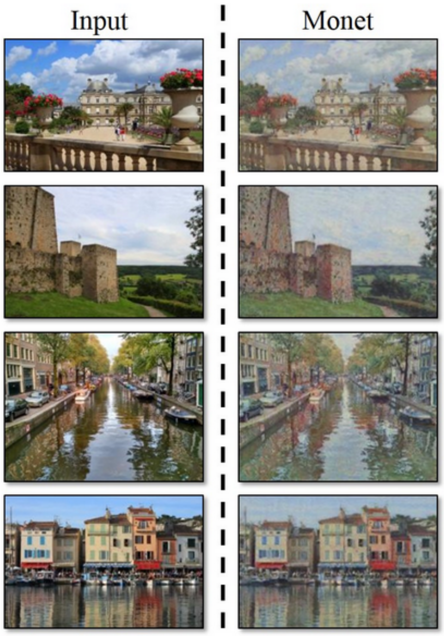

Figure 1: Examples of style transfer: photo to Monet.[5]

1 Introduction

You have learned about techniques like generative adversarial network (GAN) and image style transfer in previous lectures. In this project, you are supposed to address an interesting problem with knowledge acquired before: to transfer a photo into styles of Monet paintings, as examples shown in Fig. 1. Then, you may conduct some visualizations of features in a deep convolutional neural network (CNN) for images before and after style transfer.

2 Requirements

1 [60 points] Take part in a competition on *Kaggle*: *I'm Something of a Painter Myself*. Use the data provided in this competition to implement an appropriate algorithm to transfer a photo into Monet's styles. Then, submit your results to Kaggle and the evaluation score (measured by FID[2]) would be shown on the leaderboard. In your report, please include following items:

   - The motivation and introduction of the algorithm you used;
   - Some qualitative style transfer examples;

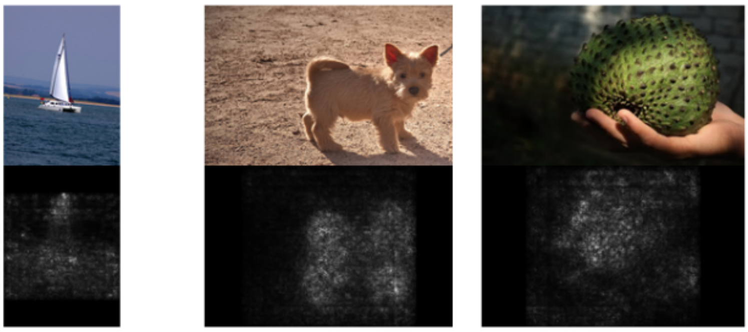

Figure 2: Examples of Saliency results.[4]

- FID score shown on the Kaggle leaderboard.

These items would be taken into consideration for the grading of your project.

2 [40 points] Implement *Saliency via Backprop* algorithm[4] you learned in the lectures. Then, evaluate the algorithm with images and class labels we provide. They are from ImageNet[1] test dataset and you may choose any public available pretrained CNN model for visualization. One possible source for PyTorch[3] users is *torch.hub*.

Note that you may need to find the correspondences between class labels and their output indexes in the pretrained model you adopt. We provide class label to index mapping for the above torch.hub models in this project. If you adopt another source, the mapping may be different. You can also use your own images as long as you can find correct correspondences.

Also, transfer the images to Monet's styles using your style transfer model in the previous part and compare the visualization difference. Define some meaningful quantitative metrics to reflect such difference. In your report, please include following items:

- A brief introduction based on your own understanding of the Saliency via Backprop algorithm;
- Some qualitative saliency results. Examples are shown in Fig. 2.
- Qualitative and quantitative comparisons on the visualization difference before and after style transfer.

3 [10 points, *Optional*] Implement *GraphCut* algorithm. Based on the saliency results in the previous part, obtain the segmentation masks,

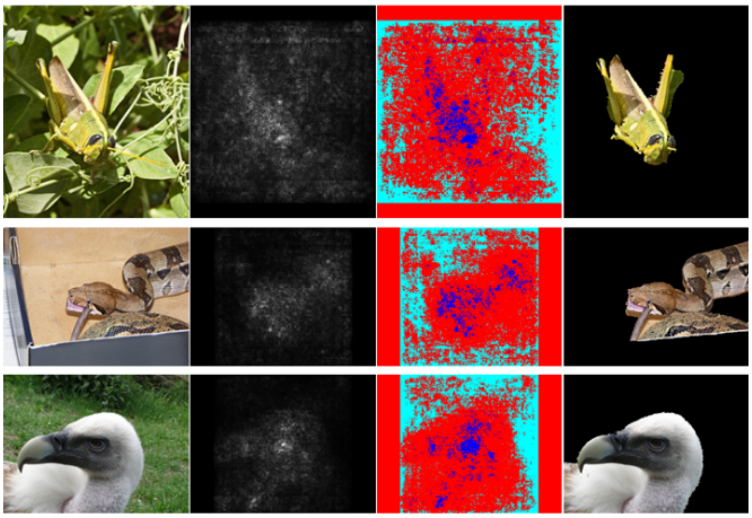

Figure 3: Examples of saliency based object segmentation.[4]

as shown in Fig. 3.

4 [10 points, *Optional*] Take effective measures to improve your style transfer algorithm, to equip it with better FID score and comparable saliency difference, or smaller saliency difference and comparable FID score. Please provide insights for your strategies and demonstrate their effectiveness with experimental study in the report.

3 Submission

Please submit your solution via *Luminus*. Put all the materials in one *zip* file named as 'ID1_ID2.zip' for a two-person group, where each ID field suggests the student ID of a group member, like *A0000000X*. Following items should be included in the zip file:

- Your report in *pdf* format including items mentioned above, as well as the responsibility taken by each member.
- Your source code for this project and make sure it is runnable.
- A *README* file describing:

  1 The environment used in your project (*e.g.*, version of python and some dependencies). Command or script used to set up the environment should also be included.

2 Command or script to run code for each part. Please clean your code and put all the steps for each part in one command or script so that it would produce the expected results for each part in an one-stop manner.

- Your trained models and input images of the qualitative results in your report. Outputs of your code should align with the ones shown in the report.

4 Tips for GPU resources

Running on GPU is more efficient for this project and it is highly recommended. If you do not have GPU resources, you may consider following options:

1 Kaggle provides free GPU hours for each contestants.

2 *Google Colab* provides free GPU access.

3 Students may apply *HPC* resources provided by NUS.

5 FAQ

Here are some frequently asked questions:

1 Q. Are some functional libraries such as graph cut lib allowed?

A. Yes. Remember to make proper citations.

2 Q. Is it allowed to refer to some tutorials or open-source code bases?

A. Yes. Remember to make proper citations. Moreover, you would **not** get a high mark if your own insight on the project is weak. For example, simply copy and paste codes or introductions from other sources.

3 Q. Is it allowed to conduct further explorations on the project requirements?

A. Definitely yes. Actually, you are encouraged to discuss issues more than those required for this project. Feel free to add some discussion to make your report more insightful.

4 Q. How to grade each group member in this group project?

A. Grading of each group member would depend on one's contribution to the whole project. By default, we assume equal contribution of all group members and we would take per capita workload into consideration. If you think it is necessary, please mention the responsibility taken by each member in your report.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[2] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[3] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[4] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[5] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.