CS F425 Deep Learning
2nd Semester 2023-24
Term Project – Task 2
Image Dehazing

The term project (task 2) for the Deep Learning course will be a team activity and is expected that you will continue with the same team as task 1. Any exception to this will require prior permission of the IC.

Description

Image dehazing is a crucial preprocessing step in computer vision that aims to remove the haze from photographs, enhancing the underlying scene's visibility and details obscured by atmospheric particles. Haze, a common atmospheric phenomenon, results from light scattering and absorption by particles like dust, smoke, and water droplets, leading to decreased contrast, color fidelity, and overall image quality. Effective dehazing facilitates a wide range of applications, from improving the performance of outdoor vision systems in autonomous vehicles and drones to enhancing the aesthetic appeal of landscape photography. Generative Adversarial Networks (GANs) have shown remarkable success in the image dehazing task. Students are recommended to go through this paper on Image dehazing.

Task 2

Data

We are providing a dataset for image dehazing consisting of hazy and haze-free pairs. The total dataset has been divided into training, validation and test set. The training and validation sets are being shared with you, while the test set would NOT be shared with you.

The dataset is being provided to you in .zip format. Once you extract it, you will find a directory structure as follows -

```
final_dataset (train+val)
├── train
|   ├── GT (7619 files)
|   └── hazy (7619 files)
└── val
    ├── GT (2050 files)
    └── hazy (2050 files)
```

All files in a particular folder are numbered like 1.ext, 2.ext, 3.ext ..... where ext is either png or jpg.

There is a folder containing the training data, and another folder containing the validation data. The number of files in each folder is mentioned above. The GT folder refers to the ground truth images, and the hazy folder refers to the hazy images. The images in these two folders are named in correspondence

both for the training and the validation dataset. This means that the hazy image corresponding to train/GT/5792.png is train/hazy/5792.png. Similarly for the validation dataset also.

While the test dataset has not been provided to you, it will also have the same structure as the training and the validation dataset, i.e. it will also have a GT and hazy folder, and the filenames in those two folders will be like 1.ext, 2.ext, 3.ext .....

Please keep in mind that the dataset contains both .png and .jpg images, and therefore, you should write code in such a way that it can take both .png and .jpg images as input.

Task

Utilize the training and validation datasets to build an **Image dehazing GAN**.

Image dehazing can be defined as an image restoration or image quality enhancement task. Given a hazy image, we want to generate its haze-free counterpart. You should utilize a **GAN based architecture** (of your choice) to perform Image dehazing.

Performance metrics

The performance metrics are SSIM and PSNR.

Evaluation

Although the task 2 is being conducted similar to a challenge or a hackathon, the evaluation will not solely be on the performance of your model. The performance metrics mentioned above will carry some weight in the evaluation. But more importantly, you will be evaluated on the *choice* of your network design, thorough experimentation, efforts to bring in some novelty in the architecture or training process, and your understanding of what-you-did and why-you-did.

Important notes:

1. Test set will not be shared with you. The performance of your model on the test set will be calculated by us.
2. Don't use any additional training data. The training+validation dataset has been carefully curated using multiple public/private datasets, and use of additional training data can run the risk of overfitting/underfitting.
3. Use of any training hacks, data augmentation approaches, pre-training techniques etc. is not just allowed but rather encouraged. However, all such techniques must be properly documented in the final report/presentation. You should explain why they were used and what benefit you got by using them.
4. Choose your GAN architectures wisely. You should be able to explain and justify your architecture and all its design choices. You are allowed to use existing code (from Github or other places) but with proper attribution. You should also remember that you need to train/test/demo your code on the free version of Colab, and a heavy architecture might not train easily on the free version of Colab.

Submission

You need to submit a brief/concise report which provides all details of the architecture chosen by you, the training process, and any other details related to your implementation. You must explain all the training or design choices adopted by you. Report the performance (as per performance metrics) you obtained for the given task. You should also report the time for training and testing, the number of parameters in your model, the choice of hyper-parameters, and loss-accuracy curves.

Final evaluation may also involve a viva or something similar. More details will be provided in due course of time.

Deadline

23rd April 2024 (midnight)