TDT4265 Final Project Proposal
GAN SISR

Eirik Ekjord Vesterkjær, MTTK

1 Motivation and brief description

The objective of the *Single Image Super Resolution* (SISR) problem is to generate high resolution output images from low resolution input images - without losing clarity and detail. This essentially involves approximating an inverse of a downsampling operator.

In many real-life situations, high resolution images are not available. Additionally, the available images may be noisy or of low quality. If high resolution and high quality images can be generated from low resolution and/or low quality input images, conventional computer vision algorithms and neural networks might be easier to apply to said images. Some other possible applications include increasing webcam image resolution and providing virtual zoom for digital maps.

A *Generative Adversarial Network* (GAN) is a neural network architecture that consists of a *generator*, which produces an output from an input, and a *discriminator*, which tries to distinguish between the generator's output and the corresponding ground truth. GANs can be trained to perform image to image translation. In this project, I will implement a GAN architecture in order to solve the SISR problem.

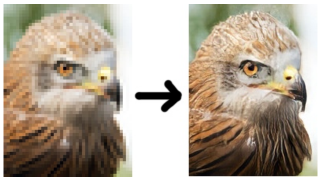

Figure 1: SISR example, images courtesy of Sajjadi et al. (*EnhanceNet*)

2 Primary goal

The main goal of this project is to,

1. design a suitable GAN architecture for solving instances of the SISR problem based on recent research,
2. train the GAN to output magnified images from input images, and
3. observe and evaluate the results through established benchmarks through testing with new (possibly noisy) data.

3 Relevant literature

Including the literature on GANs recommended by Håkon Hukkelås in the project assignment (Isola et al. (2017), Zhu et al. (2017), Karras et al. (2018), Wang et al. (2018), Gulrajani et al. (2017)), the following literature should be relevant for the project:

1. Deep Learning for SISR: A Brief Review, Yang et al. [arXiv, 2019]
2. Single-Image Super-Resolution: A Benchmark, Yang et al. [ECCV, 2014]
3. EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis, Sajjadi et al. [ICCV 2017]
4. SRGAN: Photo Realistic SISR Using a GAN, Ledig et al. [arXiv, 2017]
5. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks, Wang et al. [arXiv, 2018]

4 Datasets

This is, in my opinion, the most interesting part of this project. Seeing as images can be easily downsampled, we can in theory use virtually any image to train the network. Given a set of images, we can create a dataset for the project where,

1. the ground truths are the images themselves
2. the inputs for the GAN are downsampled versions of the ground truths (possibly with added noise)

So: there is more or less infinite amounts of possible training data available online. Additionally, extensive data augmentation can be used to further increase the amount of available data, and to ensure the network is sufficiently generalized.

I'm planning on using the *Large-scale CelebFaces Attributes* (CelebA) and *Google Open Images Extended* (GOIE) datasets for training and validation. The images will be the ground truths, and the inputs will be created by downsampling the ground truths. Since the GOIE dataset is crowd-sourced, it should have relatively varied images - and CelebA should be good for training face reconstruction. As for testing/benchmarking, I am planning on using set5 and set14 - commonly used for SISR benchmarking - and possibly some of my own photos.

5 Implementation basis

My initial idea is that both the generator and discriminator can be CNNs, but I also think it's very likely that there are some problem-specific tricks necessary for getting good performance. A more specific choice will we made when I have sufficient domain knowledge.

Both EnhanceNet and SRGAN have implementations available online. I will begin by examining their papers and implementation architectures to learn approaches to solving the SISR problem (but I am not planning on basing my project completely on existing implementations).