

Image Translation Spec

Image translation aims to translate images from one domain into another. More concretely, assume you have two different distributions of images, P_A and P_B , from which you have samples (your training data). There are two forms of this problem, which are actually quite different:

1. The supervised case, where you have a set of *pairs* A, B which you know correspond to each other. For instance, you can have pairs of colorized and greyscale images, where each pair is the same image just in the different color format.
2. The unsupervised case, where you just have a set from A and a set from B, and you don't know which image corresponds to which. For example, you can have a set of Monet paintings and a set of landscape photos, but you don't have a Monet painting of each landscape in hand.

The approaches to these specifications are different. You can choose which one you want to do for this project.

In the end, we expect a model which can translate images from one domain to the other at a reasonable resolution, something like 128 x 128. This means the model must be trained on GPU using cloud services, to which you'll have \$50 to spend. Please only run things on a single AWS account. Code written must come from the project group. *Do not* imitate a blog post or paste other's code. By next week, the following must be done:

1. Understand the following topics:
 - a. CNN's
 - b. GAN's
 - c. How to use AWS spot instances to get your training etc. done
2. Read the following papers:
 - a. <https://arxiv.org/pdf/1701.00160.pdf> A little old, but a good way to get started understanding GAN stuff.
 - b. <https://arxiv.org/abs/1611.07004> this is pix2pix, supervised translation
 - c. <https://arxiv.org/abs/1703.10593?context=cs> cycleGAN, unsupervised translation
3. Choose what domains you'll be translating between, and consequently whether you'll be working on supervised or unsupervised translation. If you need to acquire the data yourself, you need a plan to do this. (art \longleftrightarrow photos, greyscale \longleftrightarrow color, etc.) You need about a thousand members from each class.
4. Set milestones - this project has a short timeframe, but we expect you to get something you're proud of by demo day. You need to set detailed, actionable, weekly milestones to which you must hold yourself accountable.
5. Make sure you can get shit running on your laptop (tensorflow + tensorboard essentially). Make a virtual environment or a conda environment for this project.
6. Optionally, start writing a bit of code. Not necessarily the exciting parts, but just getting started.

