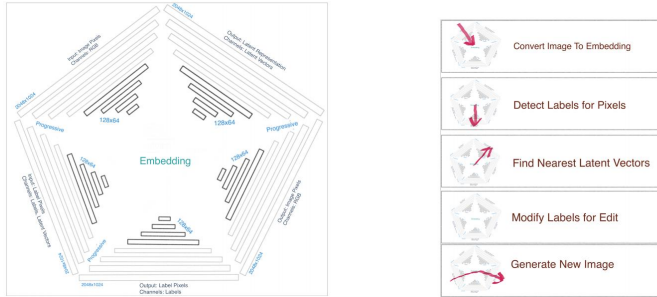


OBJECTIVES

The goal of this project is to explore using conditional Generative Adversarial Networks (GANs) for Image Editing. Primarily around removing unwanted objects from images and changing the dimensions of existing objects. GANs have been demonstrated to be able to generate high quality high resolution images. It would be interesting to see a GAN could be trained to understand an existing image and make a change to that image. So, generate a replacement portion of the image using knowledge based both of previous training and on the context of the specific image.

MODEL ARCHITECTURE

Figure 4: Model Architecture



The above figure shows the overall architecture and edit flow steps are show with red arrows over the model.

DATASET

The City-Scapes[1] dataset consists of street view images of various cities in Europe captured via a dashcam of car as well as pixel level semantic classification labelling for a subset of the data.

DISCUSSION AND RESULTS

So far, all the major model components work except for the latent decoder model. Progressive training by adding a set of layers and only retraining those appears to work relatively quickly but with the smaller size as an upper bound on quality. The generative model quality could be improved. The latent decoder model is still training slowly and so far shows an effect of color correcting aspect but since the corresponding other models with the fresh training have not caught up with past versions yet it is hard to say. If it completes training before the report is due it would be nice to try both removing objects and swapping class latent vectors between images. Future attempts to improve quality include longer training and adding layers. Such as using introducing stride 1 layers in between stride 2 layers.

GENERATOR EVALUATION



Figure 1: Generated Examples for Evaluating Generative Model (requested labels on the left)

DISCRIMINATOR EVALUATION



Figure 2: Labeling Examples for Evaluating Discriminator (ground truth in the middle)

CNN EVALUATION



Figure 3: Autoencoder Examples for Evaluating CNNs/DeCNNs (original on the right)

The CNNs and DeCNNs are inception CNN inspired blocks which have 1x1, 3x3, 5x5, and 7x7 filters as well as batch normalization and a skip connection.