Adversarial Image Reconstruction for Autonomous Underwater Robots

Cameron Fabbri, Junaed Sattar; Department of Computer Science and Engineering, University of Minnesota

Abstract

- Autonomous underwater robots often rely on visual input for decision making. However, due to many factors such as light refraction and particles in the water, the images are often times very noisy.
- We propose a method using Generative Adversarial Networks (GANs) [3] to denoise underwater images such that they can either provide better features for an algorithm, or simply provide a more visually appealing image.

Figure 1: Sample underwater images. Light refraction and blue hue cause a non-deterministic amount of noise, along with light scatter and absorption.

Related Work

Deep Colorization

Convolutional Neural Networks (CNNs) have been shown to work very well for many vision based tasks including but not limited to image colorization [5]. Given a grayscale image as input, [5] was able to show they could generate a colorized version of the image. The recently proposed Pix2Pix [7] model has shown success as well, which our work is based off of.

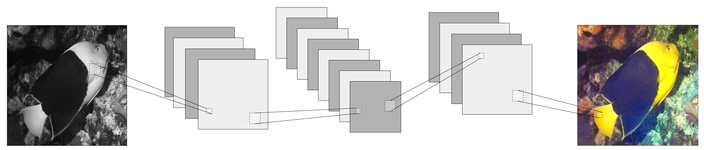

Figure 2: Image colorization example using an encoder-decoder style CNN. Images taken from [5].

Issue of Ground Truth

For the task of colorizing a black and white image, training data is easy to acquire. Any color photo can be converted to grayscale because this process is deterministic: when using the LAB colorspace, every color photo only has one associated gray image.

The same is not true for underwater photos. Underwater images face two problems:

- A ground truth image usually does not exist, given that ground truth is “What would this image look like without water?”
- Training data can not easily be generated given that a function (if one even exists) to convert a non-underwater image to appear underwater is multi-modal and nondeterministic.

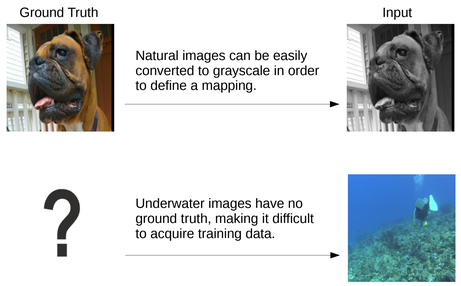

Figure 3: Examples of deterministic ground truth for colorizing black and white images, as opposed to underwater images.

Dataset Construction

To solve the issue of acquiring training data, we use the recently proposed CycleGAN [1] in order to generate a dataset of image pairs. CycleGAN learns a mapping G: X → Y such that the images sampled from G(X) are indistinguishable from the images sampled from Y, as well as a mapping F: Y → X such that the images sampled from F(Y) are indistinguishable from the images sampled from X.

We construct two domains, X and Y, where X consists of noisy underwater images taken from frames extracted from a diving video[1], and Y consists of non-underwater images as well as non-noisy underwater images taken from selected subsets from Imagenet [6]. CycleGAN is then used to generate a set of image pairs such that they appear to be underwater, as shown in Figure 3.

Figure 4: Samples of image pairs from our dataset. The top row is ground truth, while the bot row are the distorted images. Distorted images were generated using CycleGAN [5].

Method

We use Generative Adversarial Networks (GANs) to learn the task of image reconstruction. The GAN framework consists of two neural networks competing against each other. The generator *G* receives the corrupted as input and produces an image, while the discriminator attempts to differentiate between real and generated samples. The objective is shown by:

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x\sim p_{data}}[logD(x)] + \mathbb{E}_{z\sim p_z}[log(1 - D(G(z)))] \quad (1)$$

To give the network some sense of ground truth, we also consider the L1 loss as seen in (2). L1 is used as opposed to L2 because it produces less blur.

$$\mathcal{L}_{L1} = \mathbb{E}_{x\sim p_{data}}[||x - G(z)||_1] \quad (2)$$

Despite the addition of L1 to the objective function, our outputs still experienced blur. To avoid this, the Image Gradient Difference Loss (GDL) [3] is added to the objective function. The GDL directly penalizes the differences of image gradient predictions, as shown in (3), where Y is the ground truth image, and Y’ is the generated image.

$$\mathcal{L}_{GDL}(Y, Y') = \quad (3)$$
$$\sum_{i,j} ||Y_{i,j} - Y_{i-1,j}| - |Y'_{i,j} - Y'_{i-1,j}||^\alpha + ||Y_{i,j-1} - Y_{i,j}| - |Y'_{i,j-1} - Y'_{i,j}||^\alpha$$

Our final objective combines these with a weight factor. In our experiments we set $\lambda_1 = 100$ and $\lambda_2 = 1$.

$$\mathcal{L}^* = \mathcal{L}_{GAN} + \lambda_1\mathcal{L}_{L1} + \lambda_2\mathcal{L}_{GDL} \quad (4)$$

1 https://www.youtube.com/watch?v=OSdrb1XNXZI

Network Architecture

Our generator network is structured as a fully convolutional encoder decoder with skip connections, sometimes known as a “U-net”. Due to the structural similarity between the input and output, lower level features learned earlier in the network (e.g. edges and corners) that may be lost by the end of the network are in fact very important. In order to preserve these features, the activations in the encoder are concatenated onto the activations in the decoder, as shown in Figure 5.

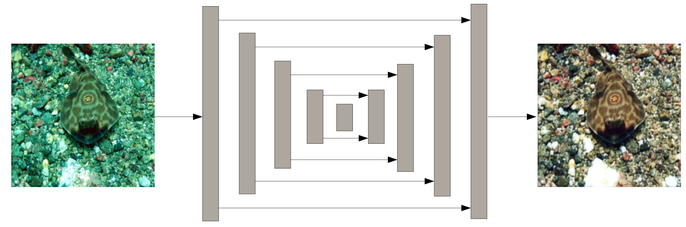

Figure 5: Our (simplified) fully convolutional U-net architecture. Activations after each layer in the encoder are concatenated onto the activations in the decoder.

Below, we show the details of our networks shown in the same format as [1]. We let Ck denote a Convolution-BatchNorm-ReLU layer with *k* filters of size 4x4 and stride 2. Convolutions in the encoder downsample by a factor of 2, while convolutions in the decoder upsample by a factor of 2. As an exception, activations in the decoder and discriminator are leaky ReLUs with slope 0.2, and the last convolution in the discriminator has a stride and kernel size of 1.

Encoder: C64-C128-C256-C512-C512-C512-C512-C512
Decoder: C512-C1024-C1024-C1024-C1024-C512-C256-C128-C3
Discriminator: C64-C128-C256-C512-C1

Results and Future Work

Figure 6 shows some test samples across multiple image classes. Our network was able to successfully recover color and lighting variations from the distorted images. Our future work involves showing how these images are able to provide better features for other underwater algorithms for mobile robots.

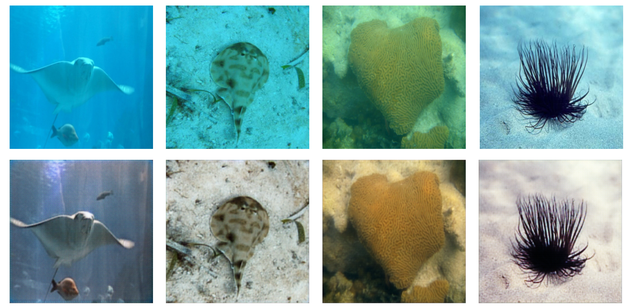

Figure 6: Some test results from our model. The top row is the input image to the network, and the bottom is corrected version. Note that the input images were not distorted by CycleGAN, they are original underwater images.

Project page and References