Fast Underwater Image Enhancement

Aryan Pandey
Department of Ocean Engineering
Indian Institute of Technology, Madras
Chennai, India
na19b030@smail.iitm.ac.in

***Abstract*—Underwater robots play an important role in oceanic geological exploration, resource exploitation, ecological research, and other fields. However, the visual perception of underwater robots is affected by various environmental factors. Due to refraction, absorption, and scattering of light by suspended particles in water, underwater images are characterized by low contrast, blurred details, and color distortion. The hue of underwater images tends to be close to green and blue. Due to this reason, Underwater Image Enhancement has a large significance in Underwater Robotics and Ocean Engineering. The state-of-the-art for this problem has evolved from simple physics based approaches to complicated CNN and GAN architectures. Most of the current state-of-the-art methods are computationally expensive and memory intensive. This hinders their deployment on portable devices for underwater exploration tasks. In this study, I build on top of the existing Shallow-UWNet, to build a model which performs high speed inference without compromising on performance. All the code written for this is available at https://github.com/aryanpandey/Fast-Underwater-Image-Enhancement**

***Index Terms*—Image Enhancement, Underwater Exploration, Underwater Robotics, Ocean Engineering**

I. Introduction

Exploration of the mysterious world of underwater has caught the attention of researchers in recent years. Analysis of underwater imaging is extremely important for ocean resource exploration, marine ecological research, monitoring of deep-sea installations and naval military applications. Light falling on the sea surface undergoes attenuation as it reaches greater depths. The larger wavelengths are affected more when compared to the shorter wavelengths. This leads to most of the underwater images appearing Greenish-Blue in colour. This limits the applicability of the images for downstream tasks like tracking, classification and detection.

In order to handle any of the mentioned issues, the first step before any downstream tasks is not only Image Enhancement but also Image Restoration. Most of the existing methods are very generic in the process of their restoration, since they extract information without any prior knowledge about the environment.In the past few years, a variety of methods have been proposed for image enhancement tasks which are of three kinds: Non-Physical Model, Physical Models and Deep Learning Models. Non-Physical models work by improving the pixel values of the image while a physical model formulates the degradation process of the image by estimating the parameters of the model. These are still not suitable for underwater image enhancement since there were some flaws that remained in the detail and exposure processing of the image.

Deep Learning methods tend to perform better as they focus solely on the colour correction aspect of the images. The existing literature based on Convolutional Neural Networks and Generative Adverserial Networks, focus on aspects such as noise removal, contrast stretch, combined improvement with multi-information and deep learning for Image Dehazing. However, these models are computationally and memory intensive which hinders their deployment on underwater robots for Image Enhancement Tasks. In the study, I build on top of Shallow-UWNet which is a lightweight method for Image enhancement which is able to maintain performance similar to that of the state-of-the-art.

Section II of this study dives deeper into some of the existing methods for Underwater Image Enhancement. Section III talks about the Problem Statement and describes some of the common evaluation metrics used to evaluate the performance of the models used for this problem. Section IV then dives into the Approach that has been used in this study. Section V talks about the results obtained and implementation details. Finally, Section VI concludes all the findings of the study.

II. Related Work

Automatic image enhancement is a well-studied problem in the domains of computer vision, robotics, and signal processing. Classical approaches use hand-crafted filters to enforce local color constancy and improve contrast/lightness rendition. Additionally, prior knowledge or statistical assumptions about a scene (e.g., haze-lines, dark channel prior, etc.) are often utilized for global enhancements such as image deblurring, dehazing, etc.

Over the last decade, single image enhancement has made remarkable progress due to the advent of deep learning and the availability of large-scale datasets. The contemporary deep CNN-based models provide state-of-the-art performance for problems such as image colorization, color/contrast adjustment, dehazing, etc. These models learn a sequence of non-linear filters from paired training data, which provide much better performance compared to using hand-crafted filters. Moreover, the GAN-based models have shown great success for style-transfer and image-to-image translation problems.

Traditional physics-based methods use the atmospheric dehazing model to estimate the transmission and ambient light in a scene to recover true pixel intensities. Another class of

methods design a series of bilateral and trilateral filters to reduce noise and improve global contrast. In recent work, there has been a revised imaging model that accounts for the unique distortions pertaining to underwater light propagation; this contributes to a more accurate color reconstruction and overall a better approximation to the ill-posed underwater image enhancement problem. Nevertheless, these methods require scene depth (or multiple images) and optical waterbody measurements as prior.

On the other hand, several single image enhancement models based on deep adversarial and residual learning have reported inspiring results of late. However, most existing models fail to ensure fast inference on single-board robotic platforms, which limits their applicability for improving real-time visual perception. Through this study, I aim to build a framework that can retain performance of these Adverserial models while being cost and memory effective so that the deployment aspect of these models improves.

III. Problem Definition

A. Problem Statement

Underwater Images are affected by various environmental factors. Due to refraction, absorption and scattering of light by suspended particles, they are characterized by low contrast, blurred details and colour distortion. This has lead to a new area of research that aims to tackle this issue. This proves useful in improving the performance of downstream tasks such as Underwater Object Detection or Underwater Semantic Segmentation.

In this problem, the aim is to learn a map (Eq. 1) that can convert any image which has either a low contrast, a colour distortion, blurred details or a combination of the three into a High resolution, High contrast Image in which colour details would match that of the objects if they were placed above the water surface. In this problem we try to learn,

$$f : B \rightarrow H \tag{1}$$

where B represents the domain consisting of Images that have a low resolution, colour distortion or low contrast and H represents the domain consisting of the corresponding High Resolution or High Contrast or Colour Corrected Images.

In this Study, I have used the EUVP dataset (Enhancement of Underwater Visual Perception) for the training, evaluation and testing of the network. Some samples from the dataset are shown in Fig.1. The First Column shows images belonging to the domain B and the second column shows images belonging to the domain H. In the first row, we can see that the task at hand here is to improve the resolution of the image as there are some blurring effects, whereas in the second and third row, the main task at hand is to improve the colour contrast and do some colour correction.

B. Evaluation Metrics

For this particular problem statement, there are some widely used metrics which I have also evaluated my network on. The metrics used are Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure(SSIM). These metrics quantify the structural similarity and reconstruction quality of the generated High Resolution Image with the True High Resolution Image.

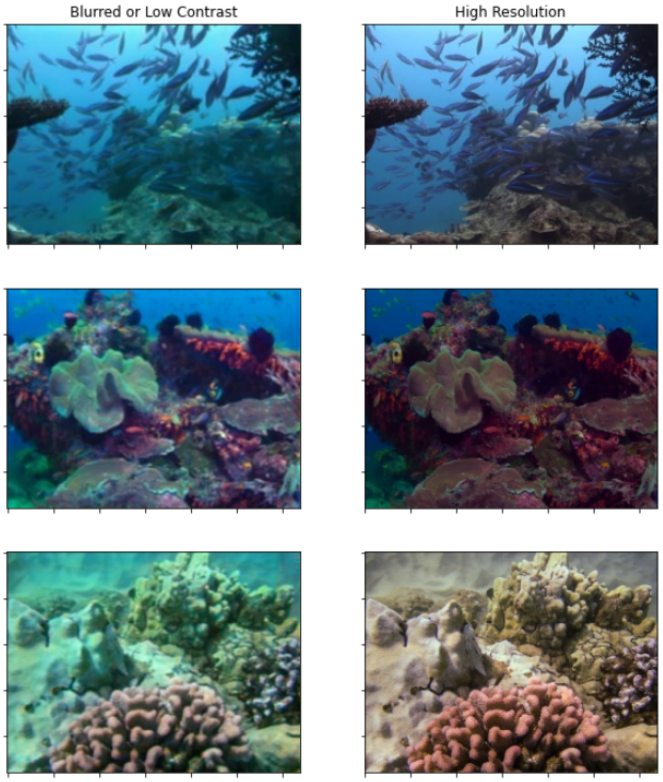

Fig. 1. Samples From the Dataset: The left side of this figure shows images that are either blurred or have a low contrast whereas the right side show the corresponding image with a High Resolution or High Contrast.

The PSNR of two Images is calculated as shown in Eq.2.

$$PSNR = 10log_{10}\frac{R^2}{MSE} \tag{2}$$

where R = 255 and MSE is the pixel-wise Mean Squared Error between the generated High Resolution Image and the Ground Truth High Resolution Image

The SSIM is calculated for two sets of images and is given as shown in Eq.3.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{3}$$

Where μ_x is the average of x, μ_y is the average of y, σ_x^2 is the variance of x, σ_y^2 is the variance of y, σ_{xy} is the covariance of x and y and c_1 and c_2 are stabilising constants for weak numerators or denominators.

While there are a huge range of metrics that one could use for this task, like the UIQM (Underwater Image Quality Measure) or Model Compression, I stick to these two metrics along with the inference times as a way of measuring the capacity and deployability of the model.

IV. Method

A. Network Architecture

Fig.2 shows the architecture diagram of the proposed network. The model comprises of a fully connected Convolution Network connected to three densely connected convolutional blocks in series. The Global Skip Connection is responsible for concatenating the input image to the output of each block. The input to the model is a 256x256 RGB underwater image. The input image is passed through the first layer of convolution with kernel size 3x3 to generate 64 feature maps, followed by a ReLU activation layer, then chained with three convolution blocks. A final convolution layer with 3 kernels generates the enhanced underwater image.

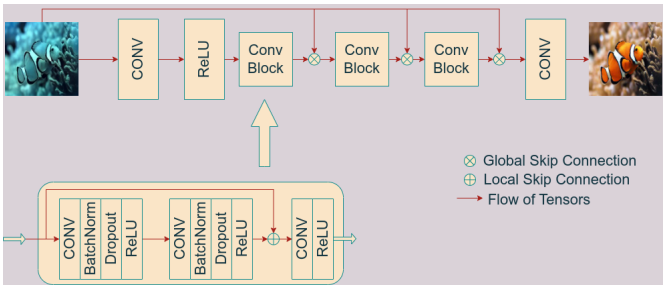

Fig. 2. Network Architecture Representation

1) Convolutional Blocks: The ConvBlocks consists of two sets of convolution layers, each followed by a batchnorm, a dropout and a ReLU activation function. There exists a skip connection from the input to the ConvBlock to the output of the second layer where the pixel values are added and divided by $\sqrt{2}$ to maintain the range of the norm of the tensors. The output is then passed through another set of Conv-Relu pair which facilitates concatenation of the raw image from the skip connection. This kind of a network structure acts as a deterrent for overfitting which means better generalisation of the network.

2) Skip Connections: Skip Connections are one of the most useful forms of information propagation. If the model encounters the issue of vanishing gradients while training, a skip connection would facilitate gradient backpropagation and would place a higher weightage on the layer where the skip connection originates from.

In this network, unlike Shallow-UWNet, there are two kinds of skip connections. The first is a Global Skip Connection in which the input image is concatenated to the output of each ConvBlock. In general this kind of a skip connection ensures that there is structural integrity in the image. The other kind of skip connection is the local skip connection in which the input to the ConvBlock is averaged with the output of the second layer of the ConvBlock. The kind of skip connection ensure that there is better feature understanding. The benefits of having this kind of a skip connection will be discussed in Section V.

B. Network Loss

The model is trained using multiple loss functions to preserve sharpness of edges in the image and impose structural and texture similarity of the generated High Resolution Image. It is calculated using the two loss components:

1) MSE Loss: The pixel-wise mean squared error (MSE) loss computes the sum of squared differences between the generated High Resolution Image I, and the High Resolution ground truth image, I* as shown in Eq.4

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(I_i - I_i^*)^2 \tag{4}$$

2) VGG Perpetual Loss: The perpetual loss is defined based on the ReLU activation of the pre-trained 19 layers VGG Network. The generated High Resolution Image and the High Resolution ground truth image are passed through the pretrained VGG network to get the feature representations. The perceptual loss is then calculated as the distance between the feature representations of the generated High Resolution Image, I, and the high Resolution ground truth clear image, I*, which is denoted by L_{VGG}

Finally the total loss is calculated as a simple sum of the two above stated losses as shown in Eq.5

$$L_{TOTAL} = L_{MSE} + L_{VGG} \tag{5}$$

C. Dataset and Training Details

All the training, validation and testing has been done on the EUVP dataset. The EUVP Dataset (Enhancement of Underwater Visual Perception) is a large collection of 10K paired and 25K unpaired images of poor and good perceptual quality. For the training of the model, I have used the Paired images alone. The Paired images were generated by distorting real world images using an underwater distortion model based on CycleGAN.

The images in this dataset were collected using a variety of cameras such as GoPros, low light USB etc during oceanic explorations under various visibility conditions and the corresponding paired images are generated using CycleGAN. Since the EUVP dataset captures locations and perceptual quality diversity, it helps the model is being more generalised to unseen images. The model is trained on 9663 images while 1705 are used for validation. The input images are of various resolutions 800 × 600, 640 × 480, 256 × 256, and 224 × 224 which are resized to 256 ×256 before training the model.

The model was trained using ADAM optimizer with learning rate set to 0.0002 and layers dropout set to 0.2. The batch size is set to 8, although a higher batch size would possibly indicate better performance. It takes around 12 hours to train the model over 50 epochs. I use Pytorch as the deep learning framework on an Intel(R) Core(TM) i7-8750H CPU, 32GB RAM and an NVIDIA Quadro RTX 5000 16GB GPU. Note that this large GPU is needed mainly because of the batch size during training, while inference, even a much smaller GPU will give predictions in a small amount of time. All the testing has been done on an NVIDIA RTX 2060 6GB GPU.

V. Results

In this section I analyse the results both quantitatively and qualitatively. The proposed model is able to achieve significant performance and is able to tackle the key issues of low resolution and colour distortion.

Fig.3 depicts some of the results. The first column depicts the image in the domain B, which is the low resolution, low contrast or colour distorted image, the second column depicts the output of the model and the third column depicts the High Resolution ground truth image.

In the first row the model is able to clearly tackle the issue of the Blue-Green Contrast while maintaining the regions where the Ground Truth is Blue-Green as well. In the second row, we can see that there is a clear resolution improvement as given by the model. In the third row, we can see that a combination of these two issues has been addressed.

Fig. 3. Predictions from the Model: Column 1 shows the Low Quality Image, Column 2 shows the Image generated by the model and Column 3 shows the High Resolution ground truth image

Another interesting observation that one can make regarding the results is that the model has in some scenarios learnt better representations of an image than what is offered by the ground truth image. I associate this better learned representations to the VGG Perpetual Loss.

Fig.4 shows one such scenario where the model generated output is of a higher quality than the ground truth itself. Since the VGG has been trained on the Imagenet and has a deep understanding of objects as they would be above the surface of water, adding in this component of the loss has led to the model learning a better representation of objects if they were kept above the water surface.

I run inference on 515 testing images which are a part of the EUVP dataset. All the inference has been done on an NVIDIA RTX 2060 6GB GPU. On this system it takes 13.6 seconds to run inference on all the 515 images, leading to an average of 0.0264 seconds per image, which is 20 times faster than WaterNet, 8 times faster than Deep SESR and 9 times faster than FUnIE-GAN.

Fig. 4. VGG Loss helps the model learn true representations better than what is offered by the High Resolution ground truth images

The mean PSNR that is obtained by the model on the test 515 images is 10.94. This beats WaterNet which has a PSNR of 9.128 which is the best performing state of the art model on this test set.

Addition of the local skip connections results in better learnt features in the initial layers of the network. Due to this addition of local skip connections, we can reduce the depth of the model by one ConvBlock (thus resulting in just 2 ConvBlocks and compared to the original 3 ConvBlocks) before training and still maintain similar performance with this smaller trained network.

VI. Conclusion

The problem of Image Enhancement for Underwater Images is one that is best tackled with the help of Convolutional Neural Networks. The proposed model maintains performance while being 12 times faster on inference speed when compared to other state of the art models. Moreover, it has good generalisation capacity due to the diverse set of images that it has been trained on, which emphasizes its real world application.

It is also observed that in addition to high speed inference, in certain scenarios there is a better learnt representation than what is offered by the ground truth images. Moreover, the addition of local skip connections, gives us better learnt representations in the initial layers, which enables us to maintain performance even after dropping a part of the network.