SAPIENZA
UNIVERSITÀ DI ROMA

Underwater Image Enhancement and Restoration using Adaptive Adversarial Learning and Superpixel-based Optimization

Facoltà di Ingegneria dell'Informazione, Informatica e Statistica

Corso di Laurea Magistrale in Computer Science

Candidate

Chiara Giacanelli

ID number 1801145

Chiara Giacanelli

Thesis Advisor

Prof. Danilo Avola

Danilo Avola

Co-Advisor

Prof. Daniele Pannone

Academic Year 2023/2024

Underwater Image Enhancement and Restoration using Adaptive Adversarial Learning and Superpixel-based Optimization
Master's thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: giacanelli.1801145@studenti.uniroma1.it

Abstract

The underwater environment presents unique challenges for imaging systems due to factors such as light attenuation, color distortion, and particulate matter scattering. In contrast to the conventional approach of addressing enhancement and restoration as separate tasks in underwater image processing, the following thesis leverages innovative deep learning techniques in order to obtain an end-to-end framework capable of achieving both underwater image enhancement and restoration:

- Pixel-wise enhancement has been achieved through the implementation of an adaptive conditional adversarial framework which deals with the non-uniform degradation and color-specific wavelengths through complex structures including deformable convolutions and context-based residuals;
- Restoration has been accomplished with the implementation of an innovative optimization function that estimates the atmospheric light of underwater images via Simple Linear Iterative Clustering (SLIC) algorithm and dark channel computation;

The experimental results present the positive impact of combining such techniques, and qualitative evaluations showcase the potential effectiveness of the method in real-world advanced vision applications such as object detection and segmentation for underwater exploration.

Contents

Introduction

The ocean is a huge body of saltwater that covers about 71% of Earth's surface. An estimated 97 percent of the world's water is found in the open sea. Because of this, the ocean has a considerable impact on weather, temperature, and the food supply of humans and other organisms. Despite its size and impact on the lives of every organism on Earth, the deep still remains a mystery: in fact, more than 80 percent of its surface has never been mapped, explored, or even seen by humans. According to the National Geographic Society, a far greater percentage of the surface of the moon and the planet Mars has been mapped and studied than of our own ocean floor [41]. This is due to several reasons: firstly, the extreme amount of crushing pressure caused by oceans' depths makes them really challenging to be explored. At the average ocean depth (3,800 meters), pressure on the sea floor is as much as 380 times greater than it is at the surface and in the deepest trenches it's 1,100 times higher. Additionally, the cost of deep sea exploration is enormous, and even if there are governments' organizations such as the National Oceanic and Atmospheric Administration (abbreviated as NOAA[1]), as well as universities interested in this kind of research, they don't have anywhere near the quantity of funds needed to conduct such missions. Unlike moons and planets, the undersea floor can't be mapped with radar systems because seawater tends to block satellites' radio waves. To be able to produce high-resolution images of the seafloor, experts need to employ a series of sophisticated sonar techniques, which can map an infinitesimal fragment of the ocean depths with an approximate accuracy of about 100 meters, through a much more slower process.

Nevertheless, the oceans are probably the most important part of the Earth's ecosystem. They produce between 50% to 80% of our oxygen, and use it for the health of marine organisms. Covering 70 percent of the Earth's surface, the ocean transports heat from the equator to the poles, regulating our climate and

[1]NOAA, `https://www.noaa.gov/`

weather patterns by absorbing about 30 percent of the carbon dioxide emitted by anthropogenic activities and about 90 percent of the excess heat. Moreover, many medicinal products come from aquatic organisms, including ingredients that help fight cancer, arthritis, Alzheimer's disease, cardiovascular diseases and even Covid-19. For these reasons, in recent years, researchers have invested a lot in trying to find solutions to increase the amount of deep-sea explorations: especially, in 2021 UNESCO's Intergovernmental Oceanographic Commission (IOC-UNESCO) has set the Decade 2021-2030 as the one pledged to protect the oceans and revolutionize the use of ocean science in line with the 2030 Agenda for Sustainable Development. In order to facilitate and make underwater exploration more affordable, Artificial Intelligence is being increasingly adopted in research. Deep learning and AI are able to create cheaper instrumentation that will allow everyone to monitor the environment and learn about the open sea and its depths. The most exciting applications of AI in the study of the underwater world are *image quality recovery* and *object detection*. In fact, deep learning allows AI systems to recognize patterns and make predictions based on data. This means that such algorithms can now process low-quality underwater images, revealing details that were once hidden, through image enhancement and restoration. Deep learning systems can also identify and classify marine life and underwater structures with astonishing accuracy. For example, AI can distinguish between different species of fish, coral formations, and even detect changes in marine life populations, or predict underwater geological events such as earthquakes and volcanic eruptions.

Aims and scope

The thesis' work particularly focuses on the tasks of image quality recovery in underwater environments, properly called enhancement and restoration, taking into consideration scenes in diverse lighting conditions and with diverse objects, in order to achieve generalization. Image enhancement techniques have the main purpose of improving the interpretability and perception of information in images for further visual objectives such as object detection, segmentation, object tracking or 2D pose estimation ([18], [48]), using techniques based on computer vision and/or deep neural networks. They typically involve tasks such as:

- Denoising: removing noise from images to improve clarity and quality;
- Deblurring: removing blur or restoring details in blurry images;

- Super-resolution: increasing the resolution of low-resolution images to produce higher-quality versions;
- Colorization: adding color to grayscale images or enhancing the color quality of images;
- Contrast and illumination enhancement: adjusting the contrast and brightness of images to improve visibility and detail;

In underwater environments, this process is typically addressed in two distinct ways, called underwater image enhancement and underwater image restoration. The first one involves improving the overall visual quality of underwater images by enhancing their clarity, contrast, color balance, and sharpness. In this case, the process aims to make underwater scenes more visually appealing and informative, working pixel-wise. The latter, instead, typically uses physical laws in order to recover the original state of images, measuring the factors that lead to degradation, such as water turbidity, light attenuation, and color cast. These tasks are commonly managed separately by different benchmarks and relatively little research is available on the two together, most likely due to their different objectives, different methodologies and computational complexity they require. Nevertheless, it's considered correct to state that they can be complementary, and that algorithms can exploit both their potentials in order to recover the quality of an undersea image with extremely high precision. That is the reason why, throughout this research, the main objective has been the construction of a unique framework able to handle both tasks, trying to preserve computational complexity and important image details.

Contributions and Thesis outline

Within the following study, underwater image processing will be explored both through image enhancement and restoration in a single supervised end-to-end deep learning framework. Image enhancement will be performed employing a wavelength-based edge feature extraction and RGB recovery; restoration will be carried out through the minimization of atmospheric light degradation within the images. To handle these high-dimensionality structures, the problem will be addressed by a GAN network having an encoder-decoder generator, a PatchGAN discriminator and a custom loss function.
The main contributions of the proposed work, in relation to the current literature

about underwater image enhancement and restoration, explained in Chapter **2**, can be summarized in three key points:

1. As the first novelty, the introduction of dynamic feature extraction via deformable convolution in the downsampling stages of the encoder, to expand the receptive field with adaptive shape and improve the model's transformation capability, aiming at reducing important information loss and still keeping meaningful details, for better image dehazing and super-resolution.

2. Secondly, the deep neural network contains a sequence of residual blocks with channel specific receptive field sizes, based on the different wavelengths of colors' light traversing through water. This was inspired by the work [39] which, however, processes the same inputs in different ways and then concatenates the color-specific features in the end.

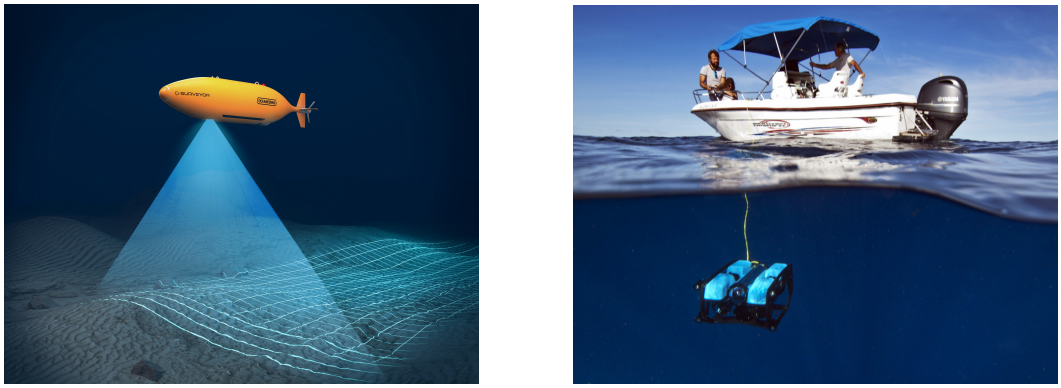
3. The proposal of a personalized loss function for underwater image restoration based on the SLIC algorithm for Superpixel generation, that estimates the atmospheric light both of the ground truth image and the generated one and minimizes the difference between the two. The atmospheric light estimation is guided by the Dark Channel Prior [28] computation.

The Thesis is structured as follows: Chapter **1** explains the fundamental concepts of underwater optical imaging, underwater enhancement and restoration, concluding with the challenges these tasks require. Chapter **2** is a in-depth revision of the state-of-the-art literature regarding the two, with particular focus on learning-based techniques; the chapter also explains the most important datasets and evaluation techniques mostly used in research. Chapter **3** is a detailed description of machine learning and deep learning models, giving particular notice to advanced architectures such as GANs. This chapter, in particular, has a dedicated section to Deformable Convolutional Networks and Residual Networks, central for the architecture of the project. Chapter **4** describes such architecture in detail, together with all the steps that helped reaching the goal. A particular section is about the SLIC algorithm and how it's used to compute the restoration-intended loss function. Chapter **5** focuses on the evaluation of the system, with different metrics and results. Eventually, the Conclusions summarize the key findings of this work, discussing its contributions to the SOTA of underwater image enhancement techniques and also the potential future improvements.

Chapter 1

Enhancement of subsea images

Research on underwater image enhancement has increased significantly over the last decade, since there has been an extensive necessity of exploring deep-sea ecosystems in order to discover and study submerged objects and cities, coral reef systems and analyze how the seafloor is mapped. In fact, acquisition of clear underwater images is of great importance for ocean engineering and ocean research where Autonomous and Remotely Operated Underwater Vehicles (ROVs and AUVs) are widely used to explore and interact with marine environments [4]. Their usage is fundamental, since the ocean's depth limits people's ability to explore the it. Unassisted, humans can't dive to profound depths, as the water's pressure and weight quickly become too much and the risk of physical injury to divers becomes more likely already starting at depths around 40 meters [10].
The difference between AUVs and ROVs is that the former conduct missions without human intervention, which the second need instead. When a mission is complete, the AUV will return to a pre-programmed location where the data can be downloaded and processed. They are also commonly known as *uncrewed underwater vehicles*. Their main advantage is the freedom: not being linked or controlled by any wire they can dive freely and deeply in hidden spots. They can further amplify data collection: an AUV is pre-programmed, meaning that crew members can lower it into the water at a determined point, and it will go off and complete its tasks without supervision. While the AUV is independently operating, the people aboard the ship can focus on other aspects of data collection or project specialties, allowing them to accomplish more in less time. On the other hand, AUVs don't work well everywhere. Interference from other ships can throw them off-course. There's also a higher risk of an AUV crashing into another ship or even another AUV in busy areas. Since they don't need to stay tethered to a ship, an AUV can struggle in a part of the



(a) Images of an AUV (left) and a ROV (right)

water with strong currents.

ROVs, instead, are submarines that a researcher operates from above the water's surface, being attached to a controller by a tether. The tether carries electrical power and/or signals to the surface so that the pilot can control the vehicle and see the camera. They are also known as *underwater drones* or *underwater robots*. The main advantage of ROVs is that they can send real time data to ship through the tether connection: in this way, operators can take action immediately when needed and are in full control of the vehicle. The tethered connection can also help in crowded areas (in oceanography, crowded areas are considered parts of the ocean where there might be a lot of other ship activity or other research activity) and in parts of the ocean where a strong current is present. The real-time tethered connection decreases the chances that interference will confuse the ROV or send it off course.

ROVs are better suited when researchers need to gather materials from the water, such as sediment, rocks or other ones. In this case, the vehicles are attached with a sediment sampler, which clusters sediment from tanks and seafloors to bring to the surface for further testing and analysis. An AUV, instead, is preferred when there's the need of capturing images or making a map of the ocean floor.

Striving to perform an in-depth research of the ocean environment has led to the necessity for these vehicles to be modelled with high-quality imaging system for effectively investigating underwater. This is due to the fact that underwater images, in general, suffer from low contrast and high color distortions caused by the non-uniform attenuation of light as it propagates through the medium. Low quality of underwater images lead to the failure of computer systems which are used for visual inspection of images. Hence, it is extremely vital to develop the underwater enhancement techniques for use in sophisticated underwater imaging tasks.

Last few decades have attracted much research in the domains of underwater image restoration and enhancement. Some of the vast amount of applications in which marine imaging (with the subsequent requirement of image quality recovery) is involved are:

- Marine Biology and Oceanography: species identification and behavior studies help researchers study the attitude, distribution, and interactions of marine organisms (for instance, *fish stock assessment* is the estimation and and monitoring of fish populations' movements and behaviors for sustainable fisheries management); benthic habitat mapping, in which high-resolution imaging is used to survey and map the ocean floor, allows scientists to study the composition and distribution of habitats.

- Environmental Monitoring and Pollution Assessment: imaging can be used to assess the impact of pollution on marine ecosystems, as well as the health and condition of coral reefs, which are important indicators of overall marine ecosystem well-being. Especially nowdays, where the impact of global warming is one of the main topics (since it's the cause of many changes in the underwater world), monitoring the health of such environments has become essential.

- Shipwreck Exploration: marine imaging technologies are used to document and explore shipwrecks and submerged archaeological sites, preserving historical and cultural heritage. In fact, for centuries, the ocean has been the only reliable highway to transport trade goods and treasure around the globe and many of them still lie undiscovered under the sea.

- Infrastructure Inspection: used to inspect and assess the condition of underwater infrastructure such as pipelines, cables, and offshore platforms.

- Locating Lost Objects or People: underwater imaging systems can aid in the search and recovery of lost objects, wreckage, or missing persons in aquatic environments. For example, ROVs have been responsible of finding a debris field containing parts of *Titan*, the five-person submersible vessel operated by OceanGate Inc, which tragically imploded the 18th of June 2023, during a multi-day excursion along the Titanic wrecks. The usage of ROVs proved to be essential, since the U.S. Coast Guard indicated that the search and rescue mission was difficult because of the remote location, weather, darkness, sea conditions, and water temperature [7].

- Recreational Diving and Exploration: underwater cameras allow scuba divers and snorkelers to capture the beauty of marine life and share their experiences with others. This is true also for movie and documentaries industries (to be mentioned is the Oscar-winner Netflix 2020 documentary *"My Octopus Teacher"*, entirely filmed underwater[1]).

- Security and Law Enforcement: imaging technologies are employed for underwater surveillance, border control, and law enforcement activities.

1.1 Underwater optical imaging

Unlike terrestrial images, images captured by underwater sensors are generally degraded by underwater environments [23]. The quality of underwater photos is dependent on numerous aspects, such as limited range of visibility, non-uniform lighting, an unwanted signal like noise and diminishing colour.
The degradation that we perceive when observing an undersea image is mainly caused by light scattering and selective absorption in water: the two effects are caused not only by the medium itself, but also by other components existing in seawater such as dissolved organic matter or small observable floating particles. Scattering effect redirects the orientation of light propagation and brings undesired backscattered light into the optical detector, leading to low contrast and generating a haze that superimposes itself on the image. Absorption effect reduces the energies of signal light according to different wavelengths (namely wavelength-dependent absorption), which makes the image visually generating greenish or bluish colour distortion. As the imaging distance increase, the two effects become more serious [12]. In fact, objects at far distances are usually occluded by a layer of backscatter (that's basically haze), and the further the observer is from the object in the scene, the more haze will be present in the ambient scene.
The process of sea photography is called underwater optical imaging and the output is mainly dependent on two mutually exclusive groups: inherent optical properties and apparent optical properties (IOPs and AOPs) of seawater. The main IOP parameters are absorption coefficient, scattering coefficient, attenuation coefficient, and volume scattering function (VSF), in which absorption coefficient and VSF are the two most basic parameters. On the other hand, the AOPs depend on both the medium and the geometric structure of illumination, such as the ambient light field,

[1] "The Making of My Octopus Teacher: Getting the Shots", `https://seachangeproject.com/stories/the-making-of-my-octopus-teacher/?swcfpc=1`

water color, clarity and turbidity in the water column, reflectance and radiance of the water surface. They are easier to measure than IOPs.
Afterwards, there will be a brief description of IOP parameters to better understand how underwater imaging works:

- Volume scattering function $P(\Theta, \phi)$: the VSF describes the angular distribution of light scattered by the suspension of particles in a direction (Θ, ϕ) at a given wavelength. It is often expressed as a function of the angles Θ (also known as the scattering angle) and ϕ (the azimuthal angle) and it's dependent on the wavelength of the incident light. In fact, different wavelengths are scattered differently by particles, contributing to the color-dependent characteristics of underwater light scattering;

- Absorption coefficient: as mentioned before, it refers to the rate at which light is absorbed by the water as it travels through it. It is a measure of how quickly the intensity of light decreases due to absorption by water molecules and other substances dissolved in the water (in fact, absorption in pure water indicates that blue wavelengths are more sensitive to absorption than red wavelengths);

- Scattering coefficient: scatters redirect the angle of the photon (elementary particles that make up light) path: in fact, scattering is a parameter that quantifies the probability of photons changing their direction due to interactions with particles or structural features in the medium; the two types of scattering encountered in underwater photography are forward scattering and backward scattering. Deviation of light from the object to the camera leads to forward scattering, which in turn results in a blurred image. Backward scattering is due to the refection of a fraction of light to the camera by water or foating particles, before it reaches the object. This leads to a low contrast and haze-like effect on the image;

- Attenuation coefficient $\beta(\lambda)$: it refers to the rate at which the intensity of light decreases as it travels through water. It is a measure of how quickly light is absorbed and scattered in the water medium, and depends on the previous parameters, together with water turbity. The attenuation coefficient is often expressed in units of inverse meters (m^{-1}). The total $\beta(\lambda)$ value is the sum of absorption and scattering coefficients.

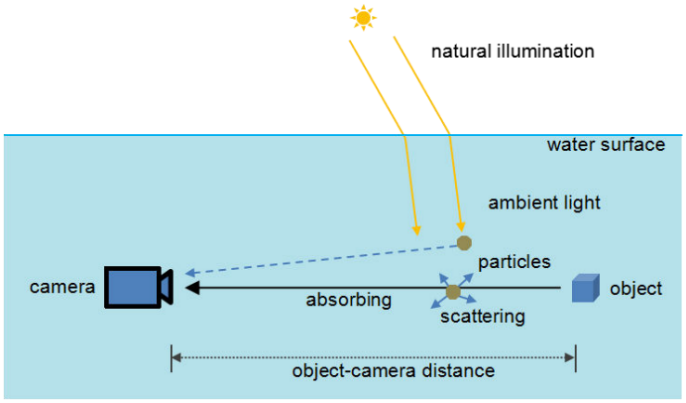


Figure 1.2. Demnostrative image of Underwater Optical Imaging

The irradiance which enters the camera consists of linear combination of three different light components namely the direct component E_d, that is the light that is refected by the object, forward-scattered component E_f, and backscatter component E_b, The total irradiance E_T is given as:

$$E_T = E_d + E_f + E_b$$

This attenuation of light is the cause of the limited visibility of objects underwater, that changes based on the depth: clarity decays at about 20m in clear water, and 5m in turbid water. In underwater environments, what we can also notice is the different kinds of light absorption based on the visible spectrum[2]: the red color is the one that's absorbed first, because it has the longest wavelength (this is why in underwater images the red channel is the one that's less visible). On the other hand, the blue colour (as well as the green one) penetrates the longest distance in water medium because of its shortest wavelength (that's 450–485 nanometers), this is why the bluish tint is predominant in underwater images.
As we can see from the table below, pure spectral colors with smaller wavelength values are the ones that have the highest frequency and photon energy. This means that they tend to be more difficult to be absorbed and, consequently, are the dominating ones in underwater photographs.

[2]The visible spectrum is the portion of the electromagnetic spectrum that is visible to the human eye. Electromagnetic radiation in this range of wavelengths is called visible light or simply light. A typical human eye will respond to wavelengths from about 380 to about 750 nanometers.

Color	Wavelength (nm)	Frequency (THz)	Photon energy (eV)
violet	380–450	670–790	2.75-3.26
blue	450–485	620-670	2.56-2.75
cyan	485–500	600-620	2.48-2.56
green	500–565	530-600	2.19-2.48
yellow	565–590	510-530	2.10-2.19
orange	590–625	480-510	1.98-2.10
red	625–750	400-480	1.65-1.98

Table 1.1. Pure spectral colors.

Absorbption of colors doesn't just depend on the spectral properties but also on water cleanliness and turbity. Water turbity is different based on the watertype. In the open ocean turbity is made of particles which are almost always drifting algae called *phytoplankton* with a well characterized spectral signature, attenuating the longer wavelengths (i.e., red colors) much faster than shorter ones, resulting in an overall bluish appearance. In the coastal oceans, instead, there are also much more optically active impurities dumped by rivers or from agricultural runoff which may dominate, causing short wavelengths to attenuate just as strongly as long ones. In Figure **1.3**, we can observe how water quality impacts the absorption of colors and how it changes the amount of attenuation coefficient (expressed in logarithmic scale) needed for the color to be visibe underwater. Types I-III are oceanic waters, while those suffixed with 'C' represent coastal waters with increasing turbidity from 1 to 9, and commonly the notion which states that water attenuates red colors faster than blue/green holds just for oceanic water types. Such classification of seawaters was derived by Jerlov [22] and it is based on the downwelling irradiance of sunlight, that is an AOP that indicates the vertical movement of surface water downward in water column caused by amount of solar radiation that penetrates it.

1.1.1 Image Formation Model

Obviously, the enhanced image exhibits more information with superior visual quality: this is particularly evident from its pixel intensity histogram, that usually shows a more homogeneous distribution of red, blue and green colours with respect to the original image. This explains the need for enhancement and restoration methods for obtaining good quality images in the field of research and real-life applications.

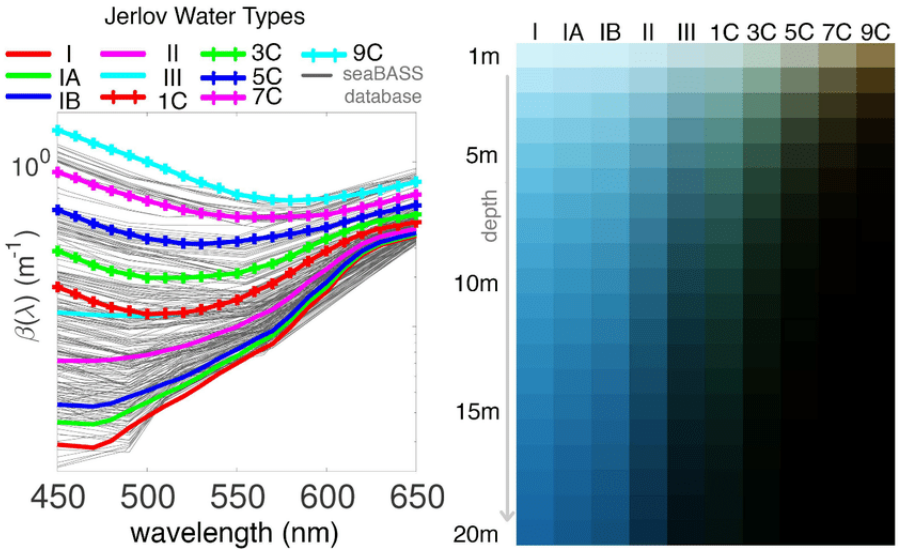


Figure 1.3. RGB simulation of the appearance of a perfect white surface viewed in 1-20m depth in different Jerlov water types (Image from: [3]).

In fact, underwater image processing can be addressed from two different points of view: as an image restoration technique or as an image enhancement method. The main difference between the two methods is that enhancement is not dependent on Image Formation Model (IFM), which restoration is dependent on instead. Before deepening the concepts of restoration and enhancement, it's necessary to describe what the IFM model is and its applications in underwater image quality recovery. The general IFM, that models the process of degradation of the atmospheric scattering model, can be formalized as follows:

$$I^c(x) = J^c(x)t(x) + B^c(x)(1 - t(x)), \qquad c \in \{r, g, b\}$$

where $I(x)$ is the observed intensity at pixel x. $J(x)$, $B(x)$ and $t(x)$ are scene radiance, background light and transmission map $\in [0, 1]$ at pixel x, respectively. The transmission map formula is: $t(x) = exp(-\beta d(x))$, where β is the attenuation coefficient and $d(x)$ is the depth of the scene. c is the color channel associated to the parameters.

This is a general model, usually adopted in the context of underwater image restoration, even if it models the outdoor haze. In fact, this model neglects the properties of underwater imaging and lighting conditions, and this makes it not always suitable for underwater scenarios. As shown in Figure **1.4**, the wavelenghts dependencies in the atmosphere are quite different with respect to the ones in the ocean environment. In fact, the lines that depict air functions are almost straight: this means that there's

not a lot of color dependency in the atmosphere, especially in haze or fog, which are in fact mostly white. When looking at the attenuation in the ocean, on the other hand, lines are curved and using the original IFM causes more errors when trying to dehaze an underwater image. For this reason, variations of this model have been proposed in order to better work according to the principles of underwater ambience and to obtain better performance, considering direct and backscattering light and estimating the medium transmissions of the three color channels, based on the different attenuation coefficient β and wavelength. One of the first innovative methods that has been introduced is called *"Sea-Thru"* [2] and it proposes a revised Image Formation Model that takes into account reflectance, ambient light, physical scattering and attenuation, and imaging angle:

$$I^c(x) = J^c e^{-\beta_c^D(V_D)z} + B_c^\infty(1 - e^{-\beta_c^B(V_B)z}), \quad c \in \{r, g, b\}$$

In this formula $-\beta_c^D(V_D)z$ and $-\beta_c^B(V_B)z$ are the attenuation and backscattering coefficients. The innovation lies also in the introduction of the V_D and V_B variables, dependencies that make the coefficients not constant in the scene, but changing with respect to the ambience and even object colors, following the principles already described of wavelenght and watertype dependency.
Many other adjusted IFMs have been introduced in the context of underwater image restoration and enhancement during the latest years: the paper *"Underwater Image Restoration Based on A New Underwater Image Formation Model"* solves the problems of low contrast and color casts of the degraded underwater image at the meantime, inspired by a new underwater Image Formation Model, which takes the fact that wavelength dependent attenuation of underwater light and color casts of underwater image into account [56]:

$$I^c(x) = L^c M^c(x) t^c(x) + L^c(1 - t^c(x)), \quad c \in \{r, g, b\}$$

where L^c is the colors of light source, $M(x)$ is the surface reflectance which represents the restored underwater image without attenuation and color casts, and $t^c(x)$ is the medium transmission. Another underwater IFM model has been proposed in [8]:

$$D^c(x) = (I^c(x) - B^c)e^{\beta_c^D d} + B^c, \quad c \in \{r, g, b\}$$

where D is the resulting dehazed underwater image, B is the ambient light and β_c^D denotes the coefficient for direct-transmission for a distance d.
It's now evident how much importance defining a correct underwater IFM has in the context of image restoration and enhancement, since the layers of backscatter

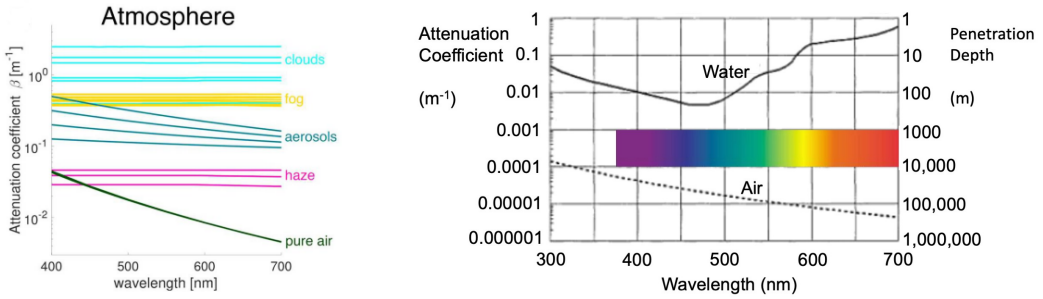


Figure 1.4. Light attenuation differences in the atmosphere and in the ocean.

and haze covering the ambience are formed in much more complex ways than what happens on the ground, and it's completely necessary to consider all these factors in order to obtain truthful reconstructions and good scene recoveries.

1.2 Image Restoration Models & Enhancement Models

As mentioned before, works in literature assess the problem of recovering the quality underwater images in two different ways: through image restoration or through image enhancement.

Image restoration aims to recover a degraded image using a model of the degradation and of the original image formation; it is essentially an inverse problem: in fact, the purpose of restoration is to deduce the parameters of the physical model and then recover the underwater images by reserved compensation processing. These methods are rigorous but they require many model parameters (like attenuation and diffusion coefficients that characterize the water turbidity) which are only scarcely known in tables and can be extremely variable. Another important parameter often used is the depth estimation of a given object in the scene.

Image enhancement, on the other hand, uses qualitative subjective criteria to produce a more visually pleasing image and it doesn't rely on any physical model for the image formation. These kinds of approaches are usually simpler and faster than deconvolution methods [38]. Underwater image restoration and enhancement models are often used in different types of applications, precisely because they work in different ways: enhancement is often used in applications where the goal is to make images more visually appealing for human observers, such as in underwater photography, tourism, or entertainment; whereas restoration is crucial in scientific and industrial applications where accurate representation of underwater scenes is required, such as marine biology, underwater archaeology, or surveillance.

There's an interesting branch of models, that's basically a combination between the two methods, and they are called *physical model-based methods*. Physical model-based methods are an interesting branch of enhancing methods, which exploit physics' laws to better understand the context of the image. To be more precise, they can be defined as an hybrid method between enhancing methods and restoring (dehazing) methods. These models regard the enhancement of an underwater image as an inverse problem, where the latent parameters of an image formation model are estimated from a given image. These methods usually follow the same pipeline:

1. Building a physical model of the degradation;
2. Estimating the unknown model parameters;
3. Addressing this inverse problem.

Many research papers locate them among the Image Restoration Methods, since these processes require physical models of degradation, which is dependent upon parameters like turbidity, time, attenuation coefficient [40]. So, even though this techniques actually lead to image enhancement, for many researchers exploiting the prior of physics is enough to consider them as such. As described in Chapter 4, the architecture of the thesis' model can be described as a physical model-based method, since it tries to estimate the atmospheric light directly from the image and tries to recover it through an optimization technique.

1.3 Challenges of Underwater Imaging

Underwater image quality recovery, as well as underwater imaging, is a complex task which requires many expedients to work well. First of all, undersea imaging systems need specialised hardware for capturing underwater photographs. Except for optical cameras, which are considered good because they are generally capable of capturing a wide range of wavelengths of light (spectral sensitivity) and have a high dynamic range (which refers to their ability to capture detail in both bright and dark areas of an image), all the other methods have their own limitations. For instance, infrared and ultraviolet imaging systems may have limited sensitivity to specific wavelengths and can be attenuated by underwater environments, sonar imaging isn't good for capturing images because it usually lacks in details and thermal imaging may have limited utility for capturing detailed visual images underwater, especially in clear water environments where there may not be significant temperature variations.

Together with this, not everyone is able to take photographies underwater, especially at great depths, making the task necessarily demand professionals to be fulfilled. The requirement of highly skilled divers for underwater exploration makes it a costly affair, because each investigation may require stand-by divers and supervisors for a single mission. Moreover, only a restricted amount of time can be spent in an underwater medium, particularly when inspections are carried out by a diver. This leads to an increase in time duration needed for the purpose of exploration. This drawback can be overcome to a great extent by utilising underwater image enhancement techniques [40].
Another important challenge of underwater imaging, that's linked to the one previously described, is the lack of large-scale labeled datasets: models of underwater image enhancement and restoration need large datasets to effectively function; but collecting and annotating these datasets can be challenging due to the costly and time-consuming nature of such underwater data collection.
As explained in previous sections, during the propagation of light through water, the optical property of water causes adverse effects to underwater imaging. This causes many consequences in underwater images. The major effects are:

- Color distortion and attenuation: underwater images typically suffer from an overwhelming color cast, such as blue or green colors that have shorter wavelengths which are absorbed more slowly. This results in a loss of color information and reduced contrast. Enhancing the colors while maintaining a realistic appearance is a significant challenge;

- Scattering and Backscatter: light scattering and backscatter, caused by suspended particles and impurities in the water, can lead to haze and reduced visibility. These effects can obscure details in underwater images.

- Non-uniform illumination: water medium is a natural filter which absorbs a considerable amount of light travelling through it. For every 10 m of depth underwater, half of the light is lost. This means that at a depth of 10 m, we just have 50 percent of the light that we had at the surface, and only 25 percent at a depth of 20 m. This can result in non-homogeneous areas of light within underwater images;

- Occlusion of objects at far distances: the layer of backscatter causes limited visibility range at great distances (the further away the object, the more haze will be present in the picture);

This is all surrounded by dynamic underwater environments. Changing conditions such as water currents, turbulence, and varying levels of natural light can indeed present a significant great challenge for image enhancement algorithms that have to adapt and consider the variety of these events, often unpredictable and all different from eachother.
Researchers and engineers are actively addressing these challenges through the development of specialized algorithms, mostly based on AI or specially-designed hardware.

Chapter 2

State of the art

2.1 Existing Methodologies

One first classification of present underwater image enhancement and restoration methods regards hardware and software based approaches ([40]).
Hardware approaches, also named *Supplementary Information-based Methods*, are based on the usage of specific technical tools and devices able to process the image as soon as having captured it. Hardware-based underwater imaging methods reduce scattering and obtain clear underwater images mainly by improving imaging sensors or systems [23]. We can distinguish four types of hardware-based approaches:

- **Polarization**[1]: in underwater images, this method leads to a significant reduction of noise. Many related works exploit the physics' polarization effects of underwater scattering taken at different levels in order to have photos at slight photometric differences. These differences serve as initial cues for a algorithms that factors out turbidity effects by reverting the "veiling" effect caused by the backscatter ([37], [36]). An interesting discovery is that marine animals use polarization for their sight underwater.

- **Range-gated imaging**: it improves signal-to-backscattering noise ratio (SBR) by rejecting backscattered light from the target irradiance. This is achieved by synchronizing the arrival of pulsed target irradiance with the gating of an intensified camera [45]. Thus, this method uses time discrimination (Reflected

[1] A polarizing filter (or polarising filter) is often placed in front of the camera lens in photography in order to darken skies, manage reflections, or suppress glare from the surface of lakes or the sea. Since reflections (and sky-light) tend to be at least partially linearly-polarized, a linear polarizer can be used to change the balance of the light in the photograph.

Image Temporal Profile) to have the desired temporal interval for underwater imaging systems. This value is called *actual RITP*, and it's basically a composition of backscattering noise, signal-scattered noise and target-reflected signal.

- **Fluorescence imaging**: this methodology is mainly used to detect microorganisms present in coral reefs. In fact, fluorescence is defined as "the emission of electromagnetic radiation, usually visible light, caused by excitation of atoms in a material, which then reemit almost immediately (within about 10-8 seconds)"[2]. A fluorescence imaging system has three main components: an excitation source that emits light in the excitation range, a camera with adequate sensitivity to detect even the weak fluorescence signal, and a barrier filter on the camera that transmits a high proportion of the fluorescence emission while blocking the excitation illumination [49].

- **Stereo imaging**: this technique is mainly used in AUVs and is designed with the aid of real-time algorithms, because it provides higher refresh rate and resolution with lower costs. Any stereoscopic image is called a stereogram. This term, refers to a pair of stereo images which can be viewed using a stereoscope. A stereogram is composed of two images since stereo imaging is based on the concept of binocular vision, making it possible to obtain depth information via stereo processing. Stereo cameras can construct a 3D structure of the underwater scene, which is much more precise than the estimation based on software.

Hardware-based methods are a good choice when having specific elements to detect underwater, especially in static situations and when being at the disposal of professional high-tech devices. On the other hand, nowdays, single underwater image enhancement has been proven to be more suitable for challenging situations such as dynamic scenes, and thus, has gained extensive attention. Another element to specify is since most experiments are conducted only in the water tank, the central challenge for hardware-based methods is the application in a real-world underwater environment, where turbidity is the main adverse element when considering underwater imaging.
Software based approaches can be classified into two types namely known as Non-physical Model-based Methods and Data-driven Methods (also called *Learning-based*

[2]Definition taken by the Britannica Encyclopedia: `https://www.britannica.com/science/fluorescence`.

Methods). The first approach deals with the manipulation of pixel values for visual enhancement of underwater images. This technique mainly works on contrast enhancement, colour correction and hybrid methods. Whilst, Data-driven Methods deal with modern Machine Learning algorithms, and they gained a special interest among researchers over the past few years.

2.1.1 Non-physical model-based methods

Non-physical model-based methods aim at modifying the visual quality of an underwater image by adjusting the pixel values without relying on any natural law, in fact these methods don't consider the physical process of underwater image degradation for the purpose. They can be defined as *RGB-based* methods, since they work mainly on the RGB and HSV color spaces.
Notwithstanding the fact that recent studies are more focused on AI techniques for UIE, we can identify three major cathegories of non-physical methods: contrast enhancement techniques, color correction techniques and retinex based techniques.
Let's first give a definition of what contrast is. Contrast is formed by the disparity in luminance refected from two adjacent planes; it's also defined as the deviation in visual properties that makes an object distinguishable from other entities and the backdrop scene. In fact, we can distinguish a high-contrast image from a low one by analyzig their histogram: for the high contrast, the image histogram should span the entire dynamic range, because the intensity rates are located along the whole range of color values. For this reason, the contrast has to be optimized for representing all the details in the input image.
Recent contrast enhancement techniques exploit DCP-based techniques to improve the visibility of low-lighting underwater images using local contrast information as a guiding parameter [28]. DCP stands for "Dark Channel Prior", and it's an assumption introduced by Kaiming He et al. in the 2009 paper *"Single Image Haze Removal Using Dark Channel Prior"*. DCP states that "non-sky regions in an image have, inside a patch of given size, at least one pixel with very low intensity in one of the color channels". This technique was used by the authors to dehaze on-ground images, and for this reason it was deemed to be appropriate to be used for underwater images in the study [28].
Color-correction techniques aim at fixing the histogram distributions of underwater images. In fact, undersea pictures tend to have a predominance of green and blue shades of color (especially the ones taken at high depth), with the consequence that the mean of the red channel is more insignificant than that of the green channel. Also,

the histogram distribution ranges of RGB channels does not cover the range from [0,255]. Color-correction techniques are frequently hybrid methods along with the contrast ones, like in the 2021 *"Color correction and adaptive contrast enhancement for underwater image enhancement"* [58]. This technique removes the color distortion of output images, takes different attenuation rates of different color channels, and designs an adaptive contrast enhancement strategy to improve the contrast of the output image.

The recent 2022 study of Jingchun Zhou et al. proposes an automatic underwater image color correction (AUCC) method based on the image patch ([60]) and depth information. The AUCC method recognizes that the depth from the object to the camera is unified in a patch of the underwater image, which calculates the color-restored factors based on depth.

Another line of research tries to enhance underwater images based on the Retinex Model[3]. This model, stated by E. H. Land and J. J. McCann, in their 1971 work *"Lightness and retinex theory"*, states that the color of an object is determined by its reflectance, that is the proportion of light of different wavelenghts that a surface reflects. In Figure 2.1, it's represented by the *L(x,y)* component.

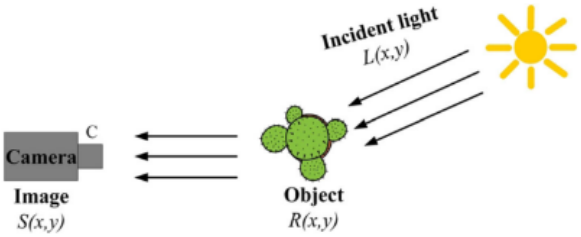


Figure 2.1. Representation of the Retinex Model

The purpose of Retinex is to eliminate the influence of the illumination component *L(x, y)*, and obtain the actual appearance *R(x, y)* of the object. This model has been used for low-light enhancement ([27], [47]) in presence of an intense noise. Retinex Model has also been used in underwater environments to restore colors, after applying the CLAHE Adaptive Histogram Equalization to the image, which limits the noise and enhances the degraded components of the underwater image at the cost of image blurring ([14]). Both the cited works lack of computational efficiency. A very recent study of 2022 also uses CLAHE along with the Retinex to enhance underwater images but, instead of applying the algorithm before, Histogram Equalization is the last step after the Retinex Model. In fact, they first add color

[3]The term "Retinex" is given by the words "Retina" and "Cortex"

correction, followed by conversion of the input image from the Red, Green and Blue color space to the Lab color space. Then, by means of the multi scale Retinex, the illumination component of the image is procured. The final enhanced image is obtained by performing CLAHE on the enhanced Red, Green and Blue image. This is intended to make the output color intensity more realistic ([42]). Despite the fact that the problem of color cast and halo effect were resolved, this technique generated not up to par images when perceived by humans.
Pixel-wise algorithms are a good choice when dealing with specific images, but the previous mentioned papers showed a lack of generalizability and precision when it comes to deal with pictures of different quality, and this is one of the reasons why recent strands of research are more tending to neural network-based models.

2.1.2 Data-driven methods

Data-driven methods, or *learning-based methods*, have recently made a huge progress in computer vision tasks, especially in image enhancement. This is due to multiple reasons, such as the generalizability power of these methods, or the robust feature learning ability. The enhanced accuracy rates associated with the deep learning algorithms exhibits the significance of this technology, clearly emphasizing the tendency for research. Encouraged by the latest success of deep learning in visual understanding and pattern recognition, researchers are working on novel underwater image synthesis algorithms, and designing of robust, data-driven solution for underwater image and video enhancement [40].
Underwater image enhancement techniques exploiting deep learning can be distinguished into CNN based methods and GAN based methods. Recently, GANs and CNNs have demonstrated powerful capabilities in various image-to-image translation tasks, including image denoising, dehazing, and superresolution.
CNN methods use Convolutional Neural Networks trained on a large amount of data to achieve the task. The first CNN implementation applied a unified learning scheme, trained with two tasks, color correction and haze removal. Formulating these two objectives together enabled learning a strong feature representation for both tasks simultaneously [52]. In order to train the model, they synthesized 200000 training images based on the physical underwater imaging model. Researchers started creating their own dataset for the purpose, as in [19], in which they introduced the new labelled SUIM dataset, along with a fully convolutional encoder-decoder architecture with skip connections between mirrored composite layers to do semantic segmentation of underwater scenes. Skip connections are introduced in order to cope

with the huge computational time required by these kind of tasks, to ensure real-time inference while achieving a reasonable segmentation performance. Also Chongyi Li et al., in their research [24] built an Underwater Image Enhancement Benchmark (UIEB) including 950 real-world underwater images, in order to train their convolutional Water-Net, introduced in the same research. This is a gated-fusion network, for the fact that the input is generated by respectively applying White Balance (WB), Histogram Equalization (HE) and Gamma Correction (GC) algorithms to an underwater image. These values are fused with the predicted confidence maps to achieve the enhanced result. Such as the just mentioned architecture, many CNN-based methods are hybrid with non-physical model-based methods, especially the ones that use the RGB color space as a prior. In 2021 researchers from the University of Electrical and Information Engineering in Tianjin built a CNN-based model using also the HSV color space in their implementation, for globally refining under-water image properties such as luminance and saturation [51]. Their network consisted of three deep-CNN blocks trained end-to-end: a RGB pixel-level block for simple and basic processing, a HSV global-adjust block that leverages neural curve layers for globally refine image property (saturation and luminance), and a attention map block for getting better underwater enhanced image through attention mechanism. CLAHE color restoration module is introduced by [59] along with a CNN defogging module to dehaze underwater images, enforcing the fact that fusion models are the state-of-the-art techniques used in the field of underwater image processing.

Despite the fact that CNNs represent a good choice in terms of obtained results, they have to be trained on a large amount of labelled data and real underwater datasets are difficult to be found. For this reason, in the latest years, trends of research are more focusing on the development of GANs as learning-based methods, and are utilizing the generative power of such networks to build their own datasets ([11], [26]). Researchers have focused on the application of GAN networks in underwater image enhancement from two directions: a conditional generative adversarial network (cGAN) and a cycle-consistent generative adversarial network (CycleGAN). UGAN network from the work [11] uses as a generator an encoder-decoder U-Net, due to the structural similarity between input and output. A generator U-Net is considered a good choice by the researchers, since many following works adopted the same structure ([18], [50], [35]). A recent work of 2023 also followed the principles of U-Net to handle large-size images, discovering that the upsampling and downsampling operations in encoders and decoders can introduce more parameters, but these

operations have more advantages when dealing with large-size images. For this reason, they introduced equally sized blocks both for the encoder and the decoder [13]. Based on experiments, the researchers found that a lightweight GAN-based model is more likely to lead to spatial inconsistency due to its limited layer channels. To avoid this problem, they included into the generator lightweight residual channel attention blocks (RCABs). Especially in very recent times, the attention mechanism is frequently been used in many other works like in the Pengfei Tang et al. *"Real-World Underwater Image Enhancement Based on Attention U-Net"* of 2023, in which they propose an underwater image enhancement benchmark based on attention U-Net which contains an attention gate mechanism that could filter invalid feature information and capture contour, local texture, and style information effectively [46]. In fact, images generated by GAN networks frequently include noise. For this reason, UMGAN includes a noise reduction network after the generator; the network's filter is based on Gaussian filtering. A 3×3 Gaussian kernel is used to convolve the generated image, which will be given as input to the discriminator afterwards [44].

2.2 Datasets

Datasets made of undersea images have been gaining attention during the past few years, due to the fact that acquiring a vast and generalizable amount of underwater pictures is a difficult task, and many researchers started creating their own synthesized data through the usage of Generative Networks, able to reproduce the undersea environment and perturbation on ordinary images.

ImageNet Dataset

ImageNet[4] is at the core of many Computer Vision models, since it contains 14,197,122 annotated images according to the WordNet hierarchy, in which each node of the hierarchy is depicted by hundreds and thousands of images. These images are currently used for object recognition or classification tasks, since ImageNet is composed of an high-quality vast amount of various data of the same object.
Cameron Fabbri et al., in their research *"Enhancing Underwater Imagery using Generative Adversarial Networks"* of 2018, selected underwater images and in-air images from several subsets of images in ImageNet and used CycleGAN to generate underwater images in pairs [11].

[4]ImageNet, `https://www.image-net.org/`

UIEB Dataset

UIEB (Underwater Image Enhancement Benchmark) is a dataset that contains 950 real underwater images. These underwater images are likely taken under natural light, artificial light, or a mixture of natural light and artificial light, so they have diverse color ranges and degrees of contrast decrease.
UIEB has been used to conduct a comprehensive study of the state-of-the-art single underwater image enhancement algorithms ranging from qualitative to quantitative evaluations by Chongyi Li et al. in their 2019 research *"An Underwater Image Enhancement Benchmark Dataset and Beyond"* [24].
As mentioned above, UIEB images have obvious characteristics of underwater image quality degradation (e.g., color casts, decreased contrast, and blurring details) and are taken in a diversity of underwater scenes. Such generalization has been shown through the construction of a Convolutional Neural Network, called Water-Net, for image enhancement.

Figure 2.2. Sampling images from UIEB

U-45 Dataset

U-45 Dataset has been introduced by Hanyu Li et al. in their 2019 research work *"A Fusion Adversarial Underwater Image Enhancement Network with a Public Test Dataset"* [25]. This is described as an effective and public underwater test dataset including the color casts, low contrast and hazelike effects of underwater degradation, used by the research in a fusion adversarial network for enhancing underwater images. It is made of 45 real underwater images, sorted into three subsets of the green, blue, and haze-like categories.

WaterGAN Dataset

WaterGAN is a large RGB-D image dataset synthesized from in-air image and depth pairings in an unsupervised pipeline [26]. Images have been created taking in consideration corresponding depth, in-air color images, and realistic underwater

images through a Generative Adversarial Network (Figure 2.3). In the paper, the generated data serves as input to a two-stage network for color correction of monocular underwater images.
WaterGAN is composed of 15 thousand images.

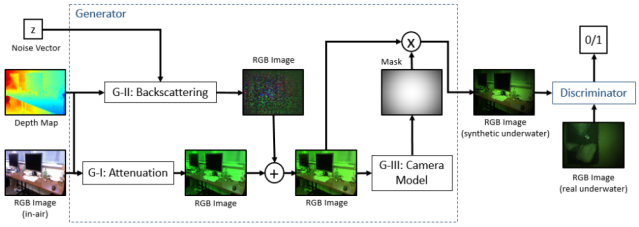


Figure 2.3. The WaterGAN process for generating synthetic images

Figure 2.4. Example images from WaterGAN Dataset.
Top row: in-air images; Bottom row: corresponding synthetic underwater images

EUVP Dataset

The EUVP (Enhancing Underwater Visual Perception) dataset contains separate sets of paired and unpaired image samples of poor and good perceptual quality to facilitate supervised training of underwater image enhancement models[5]. EUVP dataset has been introduced by the Minnesota Interactive Robotics and Vision Laboratory in the 2020 paper *"Fast Underwater Image Enhancement for Improved Visual Perception"* [18]. Data has been captured using seven different cameras, which include multiple GoPros, Aqua AUV's uEye cameras, low-light USB cameras, and Trident ROV's HD camera. The data was collected during oceanic explorations and human-robot cooperative experiments in different locations under various visibility conditions. Additionally, images extracted from a few publicly available YouTube videos are included in the dataset. The images are carefully selected to accommodate

[5]EUVP, `https://irvlab.cs.umn.edu/resources/euvp-dataset`

a wide range of natural variability (e.g., scenes, waterbody types, lighting conditions, etc.) in the data.
EUVP dataset contains over 12 thousand paired and almost 7 thousand unpaired images. Paired data has been prepared using a CycleGAN-based model to add perturbation to the images.

Figure 2.5. Paired instances from the EUVP dataset

Unpaired data has been separated into two classes: good quality data and poor quality data. These images have been used to train the CycleGAN in order to learn the domain transformation between the good and poor quality images.

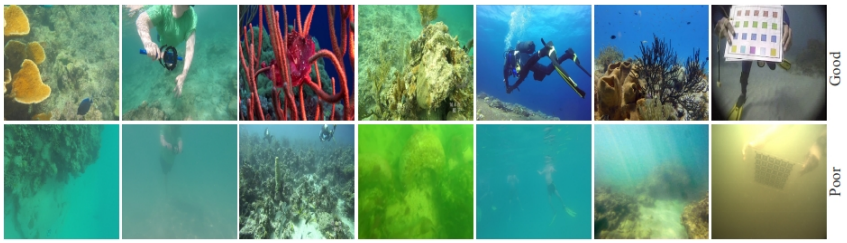


Figure 2.6. Unpaired instances from the EUVP dataset

SUIM Dataset

SUIM Dataset[6] 1525 RGB images for training and validation plus another 110 test images provided for benchmark evaluation of semantic segmentation models. This dataset has been introduced by Md Jahidul Islam et al. in their 2020 research *"Semantic Segmentation of Underwater Imagery: Dataset and Benchmark"* [19]. The images are of various spatial resolutions, e.g., 1906 × 1080, 1280 × 720, 640 × 480, and 256 × 256, and are carefully chosen from a large pool of samples collected during oceanic explorations and human-robot cooperative experiments in several locations of various water type.
The dataset contains already marked segmentation tags.

[6]SUIM, available at: `https://irvlab.cs.umn.edu/resources/suim-dataset`

Object category	RGB color	Code
Background (waterbody)	000	BW
Human divers	001	HD
Aquatic plants and sea-grass	010	PF
Wrecks or ruins	011	WR
Robots (AUVs/ROVs/instruments)	100	RO
Reefs and invertebrates	101	RI
Fish and vertebrates	110	FV
Sea-floor and rocks	111	SR

(a) Examples of SUIMD images (left) with their pixel annotations (right)

UFO-120 Dataset

The UFO-120 Dataset[7] contains 1500 for training and another 120 for testing HR natural underwater images collected from oceanic explorations in multiple locations having different water types. The saliency maps are annotated by human participants, whereas standard procedures for optical/spatial image degradation are followed to create the respective LRD samples. It was introduced in 2020 by the paper *"Simultaneous Enhancement and Super-Resolution of Underwater Imagery for Improved Visual Perception"* [20] by the Minnesota Robotics Institute.

SQUID Dataset

SQUID (Stereo Quantitative Underwater Image) Dataset[8], includes RAW images, TIF files, camera clibration files, and distance maps. The database contains 57 stereo pairs from four different sites in Israel, two in the Red Sea (representing tropical water) and two in the Mediterranean Sea (temperate water). All scenes are illuminated by natural light only and contain color charts for evaluating the accuracy of color correction.

RUIE Dataset

RUIE (Realworld-Underwater-Image-Enhancement) Dataset[9] includes more than 4 thousand images and is divided into three subsets: UCCS, UIQS and UTTS, aiming to evaluate the effectiveness of enhancement approaches from three aspects.
UCCS includes 300 images with 100 each for green images, green-blue images, and blue images; UIQS subset contains underwater images with five different qualities from high to low; UTTS contains 300 images, covering a variety of marine organisms.

[7]UFO-120, available at: `https://irvlab.cs.umn.edu/resources/ufo-120-dataset`

[8]SQUID, available at: `https://csms.haifa.ac.il/profiles/tTreibitz/datasets/ambient_forward looking/index.html`

[9]Realworld-Underwater-Image-Enhancement-RUIE-Benchmark, avalilable at: `https://github.com/dlut-dimt/Realworld-Underwater-Image-Enhancement-RUIE-Benchmark`

Additionally, the locations and types of scallops, sea cucumbers, and sea urchins are labeled manually.

Name	Year	Number of images	Annotation	Content	Source
ImageNet	2009	14.197.122	Synthetic and real underwater images	Marine life	https://www.image-net.org/
WaterGAN	2017	15.000	Synthetic images	Raw underwater and true color in-air, depth data	https://github.com/kskin/WaterGAN
SQUID	2018	122	Range, colour chart	River bed, rocks, wrecks	https://csms.haifa.ac.il/profiles/tTreibitz/datasets/ambient_forwardlooking/index.html
UIEB	2019	950	Preferred image by subjective test	Marine life and divers	https://li-chongyi.github.io/proj_benchmark.html
EUVP	2019	20.000	*Paired*: reference image *Unpaired*: quality determined by subjective test	Marine life and divers	https://irvlab.cs.umn.edu/resources/euvp-dataset
U-45	2019	45	Color casts, low contrast and haze-like effects	Marine life	https://github.com/IPNUISTlegal/underwater-test-dataset-U45-
RUIE	2019	4000	Quality determined by IQA	Marine life	https://github.com/dlut-dimt/Realworld-Underwater-Image-Enhancement-RUIE-Benchmark
SUIM	2020	1635	Marked segmentation tags	Waterbody, divers, marine life, wrecks, robots	https://irvlab.cs.umn.edu/resources/suim-dataset
UFO-120	2020	1620	Annotated saliency maps for training saliency prediction models	Marine life and divers	https://irvlab.cs.umn.edu/resources/ufo-120-dataset

Table 2.0. Summary of different underwater image datasets.

2.3 Evaluation Metrics

Many research papers conduct a subjective comparison of the effectiveness of enhancing and dehazing techniques due to the lack of a reliable objective image quality assessment (IQA) metric. This makes it difficult to fully understand the true performance of UIE algorithms, together with the fact that subjective comparison is highly time consuming, biased and more costly than a possible standard evaluation metric.
In recent years, with the high demand for automated underwater image quality evaluation, several evaluation methods have been proposed for underwater images. Objective assessment techniques, with respect to the above mentioned subjective ones, use statistical and mathematical models based on human visual system (HVS) to automatically estimate image quality.
We can distinguish two types of evaluation metrics: *full-reference image quality assessment*, in which a reference original image is available, and *reduced-reference image quality assessment*, where a reference image cannot be obtained. Many underwater image enhancement techniques are of the latest technique, since the majority of the datasets aren't composed of paired images.
The commonly-used metrics are the ones where a reference image cannot be obtained: water image quality measures (UIQM), underwater color image quality evaluation (UCIQE), and patch-based contrast quality index (PCQI), which will be described accurately in the following paragraphs.
Non task-specific evaluation metrics include:

- Measure of enhancement (EME): EME it calculates the contrast of the images and aids in the optimum selection of processing parameters.
  It is computed as:

$$EME_{m_1 m_2} = max(\frac{1}{m_1 m_2}) \sum_{l=1}^{m_1} \sum_{n=1}^{m_2} 20\, log \frac{\chi^{\omega}_{max;n,l}}{\chi^{\omega}_{min;n,l}}$$

  where $\chi_{max;n,l}$ and $\chi_{min;n,l}$ represent the maximum value and minimum value of the image within the block $\chi_{n,l}$

- Measure of enhancement by entropy (EMEE): EMEE is computed by:

$$EMEE_{m_1 m_2} = max(\frac{1}{m_1 m_2}) \sum_{l=1}^{m_1} \sum_{n=1}^{m_2} \alpha \frac{\chi_{max;n,l}(\theta)^{\alpha}}{\chi_{min;n,l}(\theta)^{\alpha}} log \frac{\chi_{max;n,l}(\theta)}{\chi_{min;n,l}(\theta)}$$

  Good image quality is indicated by high value of EMEE. $m1$ and $m2$ represent the blocks in which the image is divided.

- Mean Squared Error (MSE): MSE computes the cumulative squared error between the enhanced and the original image. Lower the MSE, better is the quality and is given as:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{i=1}^{N}(F_{(i,j)} - E_{(i,j)})^2$$

where F(i, j) is the original image, E(i, j) is the enhanced image, and M × N is image size.

- Peak-signal-to-noise ratio (PSNR): it is the measure of the peak error and computed as:

$$PSNR = 20 \cdot \log_{10}\left(\frac{\text{MAX}^2}{\text{MSE}}\right)$$

where MAX is the maximum possible pixel value of the image (often 255 for 8-bit images).

- Entropy: Entropy is a measure of information content present in the image and is given as:

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2(p(x_i))$$

where H(X) represents the entropy of the image X, n is the total number of distinct pixel values in the image and $p(x_i)$ is the probability of occurrence of pixel value x_i.

- Structure similarity index measure (SSIM): SSIM measures the similarity between original image patches and enhanced patches at locations x and y from three aspects: brightness, contrast and structure. The formula is the following:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

where x and y are the compared images, μ_x and μ_y are the means of x and y respectively, σ_x^2 and σ_y^2 are the variances of x and y respectively, σ_{xy} is the covariance of x and y and C_1 and C_2 are constants to stabilize the division with weak denominator.

2.3.1 UIQM - Underwater Image Quality Measures

UIQM comprises three underwater image attribute measures: the underwater image colorfulness measure (UICM), the underwater image sharpness measure (UISM), and the underwater image contrast measure (UIConM). Each attribute is selected

for evaluating one aspect of the underwater image degradation, and each presented attribute measure is inspired by the properties of human visual systems (HVSs) [32]. In order to better understand how the final value is obtained, let's first describe briefly the three attribute measures:

- UICM: since a good enhancement algorithm should produce a good color rendition and the HSV captures colors in the opponent plane, the two opponent color components related with chrominance RG and YB are used in the formula; more specifically, the asymmetric alpha-trimmed statistical values[10] are used for measuring underwater image colorfulness.

$$UICM = -0.0268\sqrt{\mu^2_{\alpha,RG} + \mu^2_{\alpha,YB}} + 0.1586\sqrt{\sigma^2_{\alpha,RG} + \sigma^2_{\alpha,YB}}$$

- UISM: to measure the sharpness on edges, the Sobel edge detector is first applied on each RGB color component. The resultant edge map is then multiplied with the original image to get the grayscale edge map.
  The result is multiplied with the EME (used to measure the sharpness of edges).

$$UISM = \sum_{c=1}^{3} \lambda_c EME\left(grayscale\, edge_c\right)$$

- UIConM: the contrast is measured by applying the logAMEE measure on the intensity image. The formula introduces the entropy-like operation, to the traditional Agaian measure of enhancement by entropy (AMEE), which is formulated as the average Michaelson contrast in image locations. In this formulation, PLIP operations ([33]) are used to provide nonlinear representations consistent with the human visual perceptions. Practically, lighting conditions are usually poor under the water. In such cases, the logAMEE is preferred for the reason that the log and PLIP operations put more emphasis on areas with low luminance.

$$UIConM = logAMEE(Intensity)$$

$$logAMEE = \frac{1}{k_1 k_2} \otimes \sum_{l=1}^{k_1}\sum_{l=1}^{k_2} \frac{I_{max,k,l} \Theta I_{min,k,l}}{I_{max,k,l} \oplus I_{min,k,l}} \times log \frac{I_{max,k,l} \Theta I_{min,k,l}}{I_{max,k,l} \oplus I_{min,k,l}}$$

[10] Research on natural scene colorfulness shows that colorfulness can be represented effectively with functions of image statistical values.

UIQM is computed by:

$$UIQM = c_1 \times UICM + c_2 \times UISM + c_3 \times UIConM$$

The three parameters c1, c2 and c3 are chosen depending on the type of application and sake. For instance, for color-correction applications, more weights should be applied to the UICM, while in enhancing underwater image visibilities, the contrast term UIConM and the sharpness term UISM are more significant.

2.3.2 UCIQE - Underwater Color Image Quality Evaluation

UCIQE is a linear combination of chroma, saturation, and contrast, is proposed to quantify the nonuniform color cast, blurring, and low-contrast that characterize underwater engineering and monitoring images [54].
UCIQE metric is computed as:

$$UCIQE = c_1 \times \sigma_c + c_2 \times con_l + c_3 \times \mu_s$$

where, c is the standard deviation of chroma, *conl* is the contrast of luminance and s is the average of saturation, and c1, c2, c3 are weighted coefficients.

2.3.3 PCQI - Patch-based contrast quality index

PCQI is a contrast patch-based quality model. It generates a local contrast quality map, which predicts local quality variations over space and may be employed to guide contrast enhancement algorithms. This approach represents any image patch as an N dimensional vector, in a unique and adaptive way as three conceptually independent components: mean intensity, signal strength and signal structure. PCQI is computed as:

$$PCQI(x, y) = q_i(x, y) * q_c(x, y) * q_s(x, y)$$

Where:

- $q_i(x, y)$ represents the spatial quality factor, which measures the similarity of spatial luminance patterns between the original and enhanced images.
- $q_c(x, y)$ represents the contrast quality factor, which assesses the similarity of contrast structures between the original and enhanced images.
- $q_s(x, y)$ represents the structure quality factor, which evaluates the similarity of structural luminance patterns between the original and enhanced images.

Chapter 3

Deep Learning Methods

This chapter will describe the main structures used for the project, as well as the central theoretical notions surrounding these machine learning architectures.
Machine Learning is a branch of Artificial Intelligence (AI) which focuses on the usage of labelled or unlabelled data and algorithms to imitate the way that humans learn, gradually improving its performance.
We can define three categories of Machine Learning models, based on the learning paradigm they adopt:

1. Supervised Learning: it's also known as supervised machine learning, and is defined by its use of labeled datasets to train algorithms to classify data or predict outcomes accurately.

2. Unsupervised Learning: it's also known as unsupervised machine learning, and uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. This method's ability to discover similarities and differences in information make it ideal for exploratory data analysis, cross-selling strategies, customer segmentation, and image and pattern recognition[1].

3. Semi-Supervised Learning: it's a trade-off between (1) and (2), using both labelled and unlabelled data during training. Labelled data is used to guide classification, whereas unlabelled instances (which are usually more), are used for feature extraction.

[1]From IBM: `https://www.ibm.com/topics/machine-learning`

More precisely, the learning activity is about describing the process, or model, that yields a given output y from a given input x, with some parameters Θ:

$$f\Theta(x) = y \tag{3.1}$$

The task is to solve for the parameters Θ of the function fΘ that is most likely to have produced y from x with the minimum possible error. Finding the values of the function parameters Θ is called *training*.
Mathematically speaking, the notion that describes the most this concept is the notion of *polynomial curve fitting*: in machine learning, the aim is to find the parameters of a function f that best approximate another function g that represents the given data distribution, without loosing generalization. In fact, the goal isn't to perfectly fit the input data, but to learn a function that is able to behave well also on unseen data. This is strictly related to the degree of the polynomial function and leads to two important concepts:

- Overfitting: the chosen function has a too high degree and isn't able to generalize well on unseen data. It behaves perfectly on training data.
- Underfitting: the chosen function has a too little degree and isn't able to represent well data. In this case, the model doesn't behave well both with the training data and unseen data.

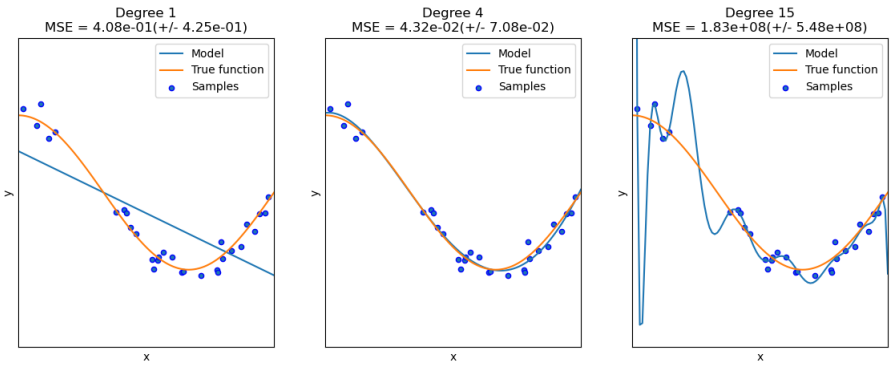


Figure 3.1. From left to right: underfitting, fitting and overfitting behaviours.

Data has a crucial role in this process. In fact, the quality and quantity of data available for training and testing play a significant role in determining the performance of a machine-learning model. Each data type has its intrinsic underlying structure that the function has to capture and the model has to find the most feasible strucures to

manage them. For example, an image is a two-dimensional representation of data, whereas a video is a sequence of frames that involves also the sequential component, and they have to be processed in different ways.
When talking about image processing, pattern recognition, natural language processing and other complex tasks, the branch that's responsible for them is called Deep Learning. Deep learning is a class of machine learning algorithms that uses multiple network layers to progressively extract higher-level features from the raw input. For example, in image processing, lower layers may identify edges, while higher layers may identify the concepts relevant to a human such as digits or letters or faces.
In deep learning, we deal with highly parametrized models, usually in the order of millions or even hundreds of millions of unknown parameters, and these models are called *deep neural networks*.

3.1 Fundamentals of Neural Networks

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are the heart of deep learning algorithms.
They are structured with layers of interconnected nodes (also called neurons, to emphasize their similarity with the structure of the human brain) and each node has an associated weight and bias value. As said before, a neural network is a composition of linear and nonlinear functions, and the simplest formula for a single-layer perceptron, that's the basic type of neural network, can be represented as follows:

$$Output = Activation(\sum_{i=1}^{n}(Input_i \times Weight_i) + Bias) \tag{3.2}$$

The weight is a floating-point number that measures the importance of the connection between two neurons (also called feature) for the final output. The higher the weight, the more important the feature. Most importantly, the weights are the *learnable parameter* by which the network makes a prediction: during the training phase of a neural network, these weights are adjusted iteratively to minimize the difference between the network's predictions and the actual outcomes, that is the error function ε. While weights determine the strength of connections between neurons, biases are constants associated with each neuron that provide a critical additional layer of flexibility to neural networks. Biases serve as a form of offset or threshold, allowing neurons to activate even when the weighted sum of their inputs is not sufficient on its own. They introduce a level of adaptability that ensures the network can learn and make predictions effectively. Another cardinal concept is the *activation*

function: these are nonlinear functions that take the weighted sum of inputs and biases as input and decide whether a neuron should be activated or not based on their output. Activation functions are beneficial because they add nonlinearities into neural networks, allowing them to learn powerful operations. In fact, if the activation functions were to be removed, the entire network could be re-factored to a simple linear operation or matrix transformation on its input, and it would no longer be capable of performing complex tasks such as image recognition. Common nonlinear activation functions include:

1. **ReLU:** $f(x) = \max(0, x)$
2. **Sigmoid:** $f(x) = \frac{1}{1+e^{-x}}$
3. **tanh:** $f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
4. **Leaky ReLU:** $f(x) = \max(0.1x, x)$
5. **ELU:** $f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha \cdot (e^{x} - 1) & \text{otherwise} \end{cases}$, where α is a hyperparameter.

These nonlinear activation functions introduce the necessary flexibility to neural networks, enabling them to learn complex mappings between inputs and outputs.
A network that's composed by both linear and nonlinear functions is called MultiLayer Perceptron (MLP), or deep feed-forward neural network. A neural network always has: an input layer, which is responsible to log in the initial values, at least one hidden layer, in which the computation of the latent features happens, and one output layer, which provides the final results.
In deep neural networks the number of hidden layers is very high in order to make the network learn intrinsic invariances of data. But how does a neural network learn? Learning consists of two important phases: the forward pass and the backpropagation.

Forward pass Forward propagation (or forward pass) refers to the calculation and storage of intermediate variables (including outputs) for a neural network in order from the input layer to the output layer.

Backpropagation Backpropagation [17] refers to the method of calculating the gradient of neural network parameters. In short, the method traverses the network in reverse order, from the output to the input layer, according to the chain rule from calculus. The algorithm stores any intermediate variables (partial derivatives) required while calculating the gradient with respect to some parameters. The gradient

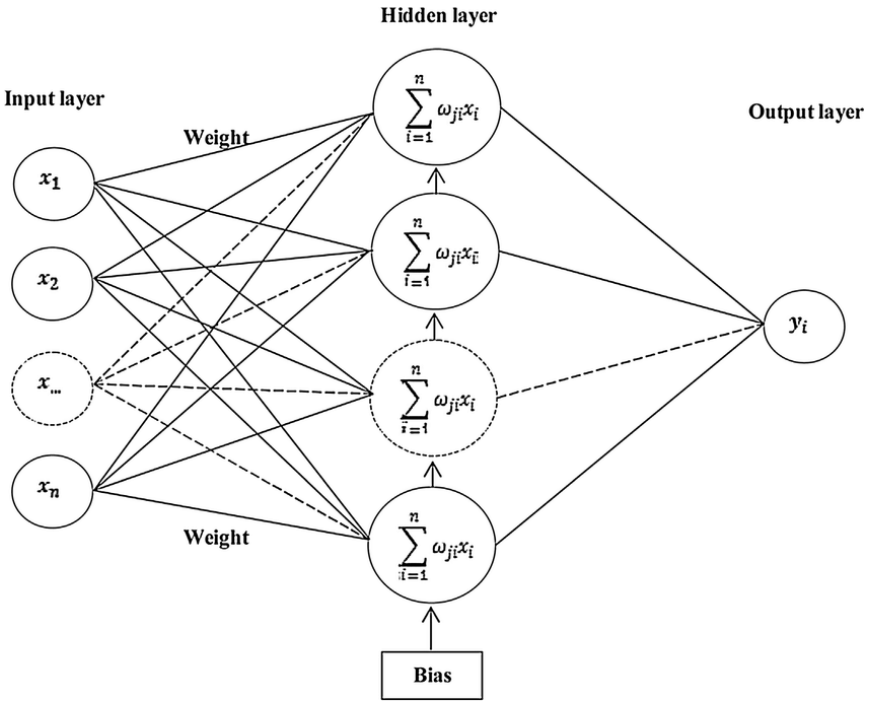


Figure 3.2. A typical architecture of a neural network.

refers to the vector $\nabla J(\Theta)$ of partial derivatives of a function J(Θ) with respect to its parameters Θ. More specifically, it represents the rate of change of the function with respect to each parameter.

Everything is controlled by the optimization of a loss function. A loss function, also known as a cost function or objective function, is a mathematical function that measures the difference between the predicted values of a model and the actual ground truth values. The purpose of a loss function in the context of machine learning, and specifically in training neural networks, is to quantify how well or poorly the model is performing on a particular task. Thanks to the gradient computation, we have information about the direction in which the function J(Θ) increases most rapidly. In the context of learning, this is called optimization, and the goal is to minimize the loss function, so the negative gradient points in the direction of the steepest decrease in the loss. Therefore, the parameters are updated in the opposite direction of the gradient.

There are many ways of parameters updating during gradient descent, such as Stochastic Gradient Descent (SGD), Momentum or Adam (Adaptive Moment Estimation).

The formula of SGD, that's the most common update rule, is given by:

$$\theta \leftarrow \theta - \alpha \cdot \nabla J_i(\theta)$$

Here, α is the learning rate, an hyperparameter that influences the length of the learning step (rate at which the algorithm makes progress) of the parameter updates.

3.2 Advanced Neural Networks Architectures

MLPs are provably universal networks, thanks to the Universal Approximation Theorem (UAT)[2]. This means that, with the enough number of units, they can approximate any function with the desired accuracy. Despite this, they aren't actually used in non-trivial deep learning tasks because they are arbitrarly complex and the number of parameters increases very rapidly. This means that they easily tend to overfitting behaviours and are very difficult to be optimized.
For this reason, multiple complex architectures have been introduced in the latest years, and this section will particularly describe the ones used in the thesis.

3.2.1 Convolutional Neural Networks

Convolutional Neural Networks [31] are a class of feed-forward neural networks that exploit the power of the Convolution operation in order to find hidden features in data. They are mostly used in visual media processing (such as pattern recognition in images or videos). CNNs are comprised of three types of layers: the convolutional layers, pooling layers and fully-connected layers. When these layers are stacked, a CNN architecture has been formed.
The convolutional layers perform an element-wise dot product between two matrices, where one matrix is the set of learnable parameters otherwise known as a kernel or filter, and the other matrix is the restricted portion of the receptive field (which is a portion of the input, always smaller than the latter). This operation is also called convolution.
The convolution operation between two functions $f(t)$ and $g(t)$ is denoted as:

$$\underbrace{(f * g)(t)}_{\text{feature map}} = \int_{-\infty}^{\infty} f(\tau) \underbrace{\cdot g(t-\tau)}_{\text{kernel}} \, d\tau$$

During the forward pass, the kernel slides across the height and width of the

[2]For any continuous function $f : A \to B$, where $A \subset R^n$ is a compact subset, and B is a subset of R^m, there exists a feedforward neural network with a single hidden layer that can approximate f arbitrarily closely.

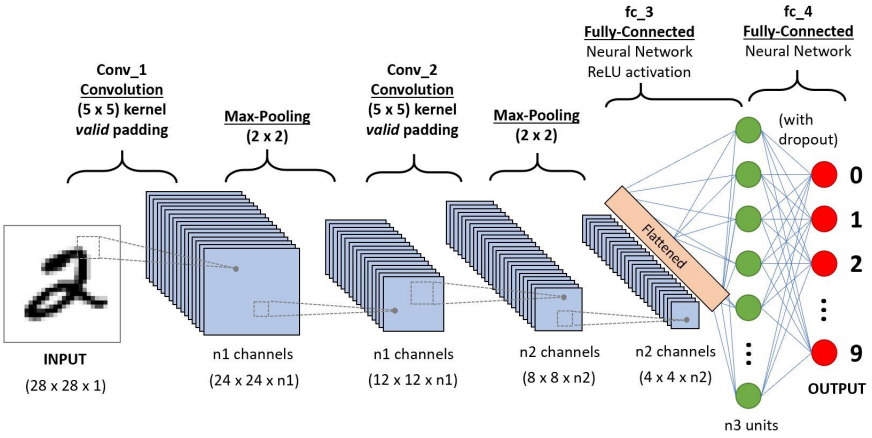


Figure 3.3. Example of a simple CNN.

image what we have called the receptive field for that specific region. This produces a two-dimensional representation of the image known as an activation or feature map and its size with respect to the input size depends on the filter size and on the stride (how much to slide the kernel over the input image). To preserve the spatial dimensions of the input and to prevent the reduction of the feature map size, extra pixels are sometimes added to the feature map (padding operation). In order to introduce nonlinearity, an activation function is always present after each convolutional layer.

The pooling layer is responsible of making the feature maps smaller and more manageable, and operates over each activation map independntly. Its objective is to make the feature maps translation invariant (not location dependent), by extracting high level features. The most common pooling functions anre max pooling or average pooling. They respectively operate over local regions of the input data and output the maximum and average value within each region. An important aspect to underline, is the invarance property of CNNs: they are invariant with respect to translation, scale and rotation and this is the main reason because they work so well with images.

Eventually, the fully-connected layer helps to map the representation between the input and the output.

The applications of CNNs nowdays are vast: they are used in image classification, object detection, semantic segmentation, image generation, gesture recognition, facial recognition, video and document analysis.

3.2.2 Residual Neural Networks

For very complex networks, it's necessary to increase the number of layers in order to better extract higher level features. But when we increase the number of layers, there is a common problem in deep learning associated with that called the Vanishing and Exploding gradient.
Vanishing gradient problem occurs during backpropagation, in which network weights are updated proportional to the gradient value after each training iteration (epoch). Depending on the type of the activation functions and network architectures, sometimes the gradient value is too small and gets gradually diminished during backpropagation to the initial layers. This prevents the network from updating its weights and also sometimes when the value is too small, the network may be completely stopped from training (updating weights) [6]. On the contrary, in some cases, the gradients keep on getting larger and larger as the backpropagation algorithm progresses (this is why we call it "exploding" gradient). When gradients explode, the weight updates during training can become so large that they cause the learning algorithm to overshoot the minima of the loss function. This can result in model parameters diverging to infinity, causing the learning process to fail[3].
In order to solve the problem of the vanishing/exploding gradient, in 2015 Microsoft Research introduced a new architecture called ResNet [16], in which the building blocks are called Residual Blocks, which are essential for the purpose of the project's network.
Residual Blocks are skip-connection blocks that learn residual functions with reference to the previous layer inputs, instead of learning unreferenced functions. This means that, in a residual block, the input to the block (let's call it x) is transformed through a series of layers, and the original input is added back to the transformed output. Mathematically, the operation within a residual block can be expressed as follows:

$$\text{Output} = F(x) + x$$

F(x) is the mapping learned by the network and x is the original input. This operation creates a *shortcut* or *skip-connection* that allows the gradient to flow directly through the block during backpropagation.
ResNet has many variants (ResNet-18, ResNet-34, ResNet-50, ResNet-101 and ResNet-152) that run on the same concept but have different numbers of layers;

[3] *"Exploding Gradient Problem"*, from the website DeepAI.org: `https://deepai.org/machine-learning-glossary-and-terms/exploding-gradient-problem`

for example, Resnet-50 is used to denote the variant that can work with 50 neural network layers.

3.2.3 U-Net

The U-Net [35] is a popular architecture for semantic segmentation. It was introduced for Biomedical Image Segmentation, and it's composed by an encoder network (also called *contracting path*) and a decoder network (also called *expansive path*).
The encoder is composed by a series of encoder blocks. Each encoder block downsamples the input image in a lower-dimensional feature map, and consists of two 3x3 convolutions, where each convolution is followed by a ReLU activation function. Downsampling happens thanks to a 2x2 max-pooling.
Skip connections are added after every encoder block, and they act as a shortcut connection that helps the indirect flow of gradients in the decoder network to the earlier layers without any degradation.
The bridge connects the encoder and the decoder network and completes the flow of information. It consists of two 3x3 convolutions, where each convolution is followed by a ReLU activation function.
The decoder, then, is used to generate a semantic segmentation mask. The goal of this network is to semantically project the higher-dimensional features (lower resolution) learnt by the encoder onto the pixel space (higher resolution) to get a dense classification. The decoder consists of transposed convolution operations and concatenation followed by regular convolution operations. The performance of the decoder is improved thanks to the skip-connections, which provide additional information to generate better semantic features.

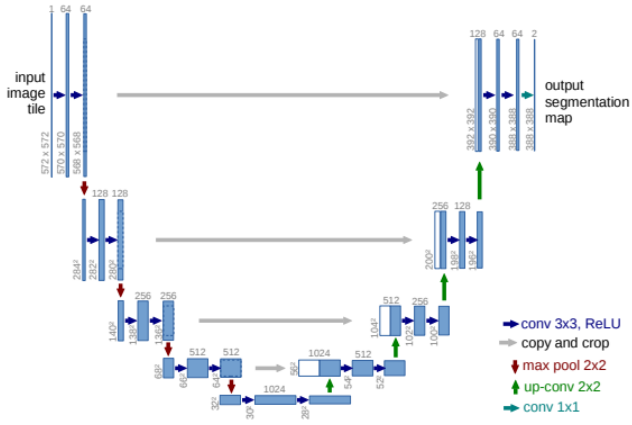


Figure 3.4. Standard UNet architecture from [35].

3.3 Deep Generative Models

Deep Generative Models are very powerful ways to learning any kind of data distribution. Differently from discriminative models, which focus on learning the boundary between different classes or categories, generative models learn a probabilistic distribution from some given training samples and therefore are able to generate new samples from the learnt distribution (essentially, they aim to discover how the data is generated). The quality of the generation will depend on how well the learnt distribution approximates the real one. Two of the most commonly used and efficient approaches are Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN).
A VAE is a specific type of Autoencoder (AE). An AE is an unsupervised encoder-decoder model trained to discover latent features of the input data:

- The encoder *E(x)* is responsible of producing a latent code z, which is a lower-dimensional representation of the input x.
- The decoder *D(E(x))* progressively decompresses z ultimately reconstructing the data back to its original, pre-encoding form.

The autoencoder is trained by minimizing the reconstruction loss:

$$l = \sum_{i=1}^{n} ||x_i - D(E(x_i))||$$

over the parameters Θ of the network.
Despite being very powerful, AEs have some limitations: they often overfit the data, leading to perfect reconstructions of training samples and very bad reconstructions of unknown data. This is due to the fact that they are strictly dependent on how the organization of the latent space by the encoder during the training process. This weakness is addressed by Variational Autoencoders, or VAE in short. The main difference between the two models is how the latent code is generated: AEs aim to learn a compressed, fixed-dimensional representation (deterministic latent space) of the input data, typically for the purpose of reconstructing the input as accurately as possible; on the other hand, VAEs aim to learn the underlying probability distribution of the training data (probabilistic latent space) so that it could easily sample new data from that learned distribution.
The probabilistic distribution form of z is decided a priori, and is denoted by P(z). For example, we can choose to represent the data distribution as a multivariate Gaussian distribution.

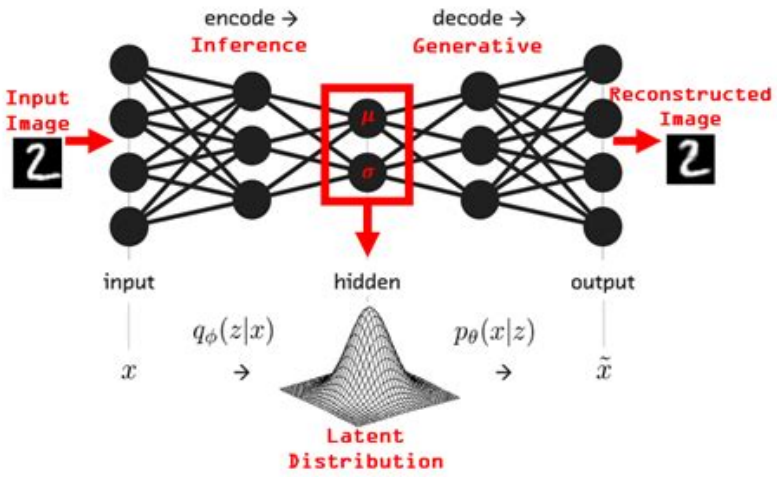


Figure 3.5. Standard VAE architecture

The final objective of a Variational Autoencoder is to maximize the Evidence Lower Bound (ELBO). The ELBO is a combination of a reconstruction term and a regularization term (Kullback-Leibler divergence).

$$\text{ELBO} = E_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \text{KL}[q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]$$

The first term is the reconstruction term, and encourages the VAE to generate reconstructions that are close to the original input x. It is essentially the expected log-likelihood of the data given the latent variables z. The second term, the KL divergence is what distinguishes VAEs from regular autoencoders. It ensures that the probabilistic encoder follows the distribution $p\Theta(z)$:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$$

where $\mathbf{0}$ is a vector of zeros and $\mathbf{I}$ is the identity covariance matrix.
Essentially, the purpose of the KL divergence term in the loss function is to make the distribution of the encoder output as close as possible to a standard multivariate normal distribution.

3.4 Generative Adversarial Networks

Generative Adversarial Networks (generally known as GANs), are a type of unsupervised deep learning architectures which exploit *adversarial learning* to generate realistic data. Adversarial learning is a general term that refers to a class of methods that, with different motivations, seek to fool models by supplying deceptive input (which is also called adversarial example). The goal of these types of models is to learn how to generate non-detectable fake data by gaining information about its intrinsic regularities or patterns.
A GAN is composed by two end-to-end networks: a Generator and a Discriminator. The Generator model takes a fixed-length random vector as input and generates a sample in the domain. The vector is drawn from randomly from a Gaussian distribution, and the vector is used to seed the generative process. We can say that the generator resembles the decoder part of the VAE. After training, points in this multidimensional vector space will correspond to points in the problem domain, forming a compressed representation of the data distribution. The goodness of the reconstruction is determined by the Discriminator, which is a discriminative network that predicts a binary class label of real or fake (generated) of the produced sample by the Generator.
Why is it called *adversarial* training? This term refers to the competition between the two networks in their objectives: in fact, the generator, parametrized by some parameters γ, is trained in such a way that a discriminator cannot distinguish between its generated sample and the real one. Simultaneously, the discriminator, parametrized by other parameters δ, is trained to maximize its ability to distinguish between fake samples and real ones. Mathematically speaking, this results in two different objective functions:

- The objective of the discriminator is the following:

$$ObjFunc_{\mathrm{D}} = \max_{\delta}(E_{x\sim p_{\mathrm{data}}}[\log D_{\delta}(x)] + E_{z\sim p_z}[\log(1 - D_{\delta}(G_{\gamma}(z)))])$$

  This function wants to maximize the probability of recognizing real samples as such (we want the value to be as close as 1 as possible) and minimize the probability of recognizing the samples generated by the Generator G, when supplied with latent codes z, as real (we want the value to be as close as 0 as possible). Here, p_{data} is the true data distribution and p_z is the latent space distribution.

- The objective of the generator is the opposite of the discriminator's:

$$ObjFunc_{\mathrm{G}} = \min_{\gamma}\max_{\delta}(E_{x\sim p_{\mathrm{data}}}[\log D_{\delta}(x)] + E_{z\sim p_{\mathrm{z}}}[\log(1 - D_{\delta}(G_{\gamma}(z)))])$$

  or, with more simplicity:

$$ObjFunc_{\mathrm{G}} = \min_{\gamma}E_{z\sim p_{\mathrm{z}}}[\log D_{\delta}(G_{\gamma}(z))]$$

  This function tries to maximize the log probability of the discriminator correctly classifying generated samples as fake.

As we can notice, the discriminator acts as a guide to help the generator learn and evolve. In fact, during training, it's always the discriminator the first network to be trained. GANs can occur in some issues, such as the vanishing gradient descent and the model collapse. The vanishing gradient problem, explained in the previous chapters, can be overcome with the usage of Wasserstein Generative Adversarial Networks (WGANs) [5], which use the Wasserstein distance (also known as Earth Mover's Distance) as the optimization criterion. Another issue that can arise is model collapse. Mode collapse happens when the generator model produces a limited set of outputs that fail to capture the full diversity of the real data distribution. In other words, the generator starts producing similar or identical samples, regardless of the input noise or latent space variations, leading to a collapse in the modes of the data distribution. Several strategies can be adopted in order to assess the problem, such as increasing the capacity of the networks, adjusting the learning rate or optimization algorithm and adding regularization techniques, such as weight decay or dropout to prevent overfitting.
Generative Adversarial Networks have found applications across various domains due to their ability to generate realistic and diverse data. Some notable applications of GANs include:

1. Image-to-Image Translation: for example, converting images taken during daylight to nighttime ambience, image colorization or sketches to realistic images;

2. Super-Resolution: to enhance the resolution and quality of images, as in the case of this dissertation;

3. Data Augmentation: to generate new realistic samples. This is especially beneficial when the original dataset is limited, and additional diverse samples are needed for training machine learning models;

4. Deepfake Generation: to create videos and images in which real faces are replaced with other identities;

5. Text-to-Image Synthesis: to generate realistic images based on textual descriptions. This has applications in creating visuals based on natural language input (such as Dall-E [34] or Midjourney [29]).

6. Anomaly Detection: to find invariances and detect anomalous behaviours in data.

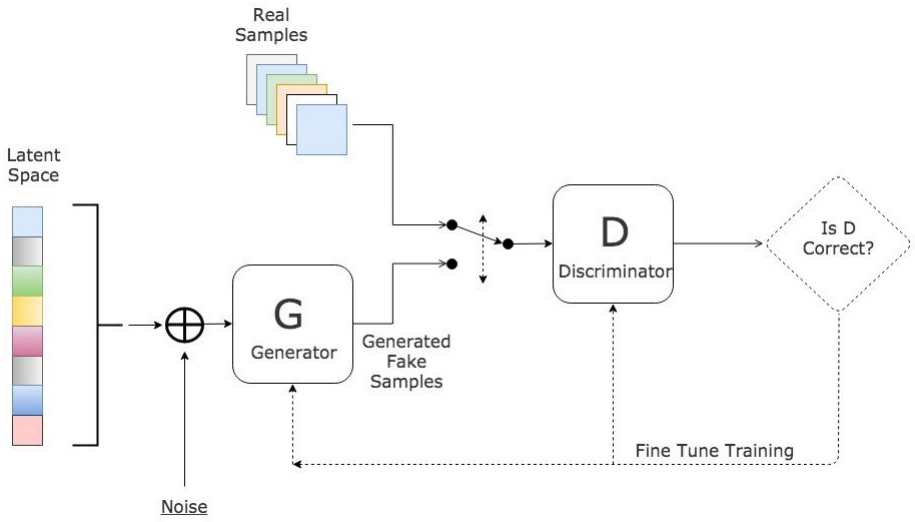


Figure 3.6. GAN structure (Image source from: `https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html`)

3.4.1 PatchGAN

PatchGAN [21] is a specific type of Discriminator network that processes data in local patches of a fixed size. So, the discriminator tries to classify whether each N x N patch in the image is real or not. The discriminator runs convolutionally across the image, averaging all responses to provide the ultimate output. PatchGAN was introduced in order to cope with L1 an L2 losses, which are good at capturing features at low-level frequencies but they fail at modelling high-frequency structures. For this reason, the discriminator was introduced to only penalize structure at the scale of patches. Such discriminator effectively models the image as a Markov random field (MRF), assuming independence between patches of pixels separated by more than a patch diameter, since the network doesn't necessarily consider the entire global context of the image.

In the context of the PatchGAN architecture, the kernel size determines the effective size of the local patches that the network processes, and the stride determines how these patches are sampled from the input image.

3.4.2 Conditional GANs

Conditional GANs [30] are a type of Generative Adversarial Networks that involve the conditional generation of images by a generator model. This means that both the generator and discriminator are conditioned on some extra information y. y could be any kind of auxiliary information, such as class labels or data from other modalities, usually represented as extra input variables.
The normal GANs learn a mapping from a random noise vector z to output image y, $G : z \rightarrow y$. The cGANs, instead, learn a mapping from an observed image x and a random noise vector z to y , $G : \{x, z\} \rightarrow y$. In the case of the project, the ground truth image serves as additional information conditioning the enhancement process. By providing the model with the ground truth image alongside the noisy input, the model is given information about what the enhanced result should ideally look like.

3.5 Deformable Convolutional Neural Networks

Deformable Convolutional Neural Networks [9] are a specific type of CNNs introduced in 2017 by Microsoft Research Asia in order to cope with the intrinsic drawbacks of such networks:

- Convolutional Neural Networks are limited to model large and unknown transformations, due to the fixed-size property of feature maps, which is dependent on the kernel size and the stride, hyperparameters for the network;
- The receptive field sizes of all activation maps in the same CNN layer are the same. This represents a great disadvantage when dealing with geometric data or images with objects with different scales and deformation in which there's the need of extracting fine-grained features. In these kind of data, it's desirable to achieve adaptive determination of scales or receptive field sizes in order to capture high-level and flexible features of such data;

This is the reason why Deformable CNNs has been introduced. These kinds of neural networks are made of two components: a deformable convolution function and a deformable ROI pooling. Both modules are light weight and operate in a 2D domain. They add small amount of parameters and computation for the offset learning. They

can readily replace their plain counterparts in deep CNNs and can be easily trained end-to-end with standard backpropagation.

3.5.1 Deformable Convolution

In deformable convolution, the regular convolutional grid R sampled over the input feature map, is enhanced with learnable offsets $\{\Delta p_n \,|\, n = 1, \ldots, N\}$, where $N = |R|$. For each location p_0 on the output feature map y, we have:

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n + \Delta p_n) \tag{3.3}$$

During training, both the convolutional kernels and the offsets are learned simoultaneously. To learn the offsets, the gradients are backpropagated through the following bilinear operations:

$$x(p) = \sum_q G(q, p) \cdot x(q) \tag{3.4}$$

$$G(q, p) = g(q_x, p_x) \cdot g(q_y, p_y) \tag{3.5}$$

where p denotes an arbitrary (fractional) location, q enumerates all integral spatial locations in the feature map x, and $G(\cdot, \cdot)$ is the learnable bilinear interpolation kernel. G is two-dimensional, and is separated into two one dimensional kernels as in 3.5.

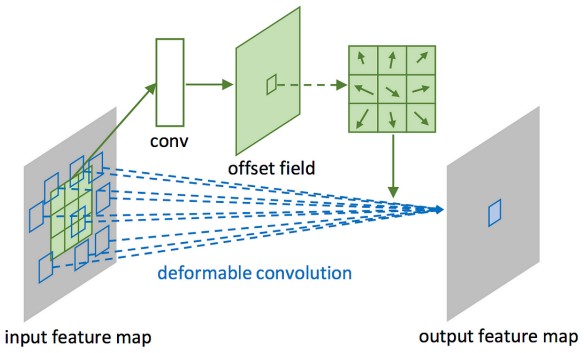


Figure 3.7. Illustration of 3 × 3 deformable convolution (Image from [9])

3.5.2 Deformable RoI Pooling

RoI (Region of Interest) pooling is an important technique used in object detection tasks. The RoI is a region from the original image: it's not a bounding box, but just a proposal for further processing. Its purpose is to perform max pooling on

inputs of nonuniform sizes to obtain fixed-size feature maps (e.g. 7×7). RoI pooling divides the RoI into kxk bins and outputs a kxk feature map y.
As in the equation of deformable convolution, in deformable RoI pooling offsets $\{\Delta p_{ij} \,|\, 0 \leq i, j < k\}$ are added to the spatial binning positions:

$$y(i,j) = \sum_{p_n \in bin(i,j)} x(p_0 + p_n + \Delta p_n)/n_{ij} \tag{3.6}$$

where n_{ij} is the number of pixels in the bin; so, the sampled values on the grids are averaged to compute the bin output. The offsets are found in this way: first of all, RoI pooling is applied to the feature map, generating pooled feature maps. The latter are put as input to a fully-connected layer, which outputs normalized offsets, which are then transformed into real offsets by element-wise product with the RoI's width and height: $\Delta p_{ij} = \gamma \cdot \hat{\Delta} p_{ij} \circ (w, h)$, where γ is a modulation term, typically equal to 0.1. The offset normalization is necessary to make the offset learning invariant to RoI size.

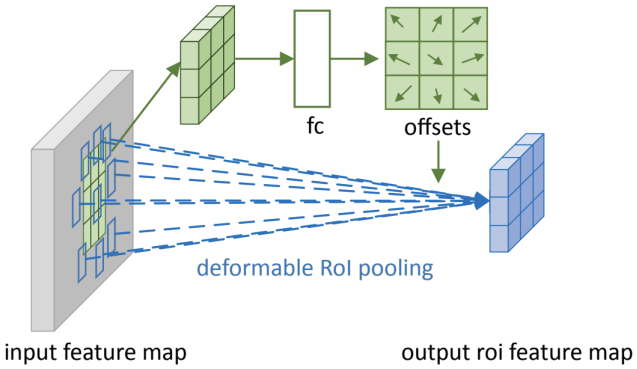


Figure 3.8. Illustration of 3 × 3 deformable RoI pooling (Image from [9])

3.5.3 Deformable ConvNets

Deformable Convolutional Networks make use of RoI pooling and Deformable layers in the same way as normal CNNs. To integrate deformable ConvNets with the state-of-theart CNN architectures, these architectures consist of two stages: feature extraction for the whole image and final segmentation or detection. The aim of such networks is to make the receptive field adaptive in order to augment the spatial sampling locations in convolution and RoI pooling. As we can see in the Figure **3.9**, when deformable convolutional layers are stacked, the resulting effect is that receptive fields are adjusted according to object's sizes and shapes. On the other

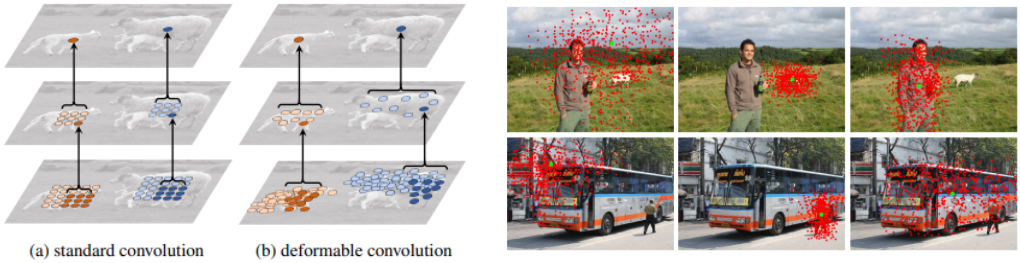


Figure 3.9. Sampling Locations of Deformable ConvNets (Images from [9])

hand, in normal convolution, the shape of the sampling locations is fixed and doesn't add any additional supervision for learning spatial transformation.
These properties can be summarized into the analysis of three central points of spatial support visualization[4] [61]:

- **Effective receptive fields**: by this term is meant that not every pixel within the receptive field of a network node contributes equally to its response. With the usage of deformable convolution, the spatial support adapts really well to the shapes of the objects in a scene. In spite of this, sometimes the receptive field of a foreground objects finds itself including background areas irrelevant for detection.

- **Effective sampling /bin locations**: these are the gradient of the network node with respect to the sampling or RoI locations, as to understand their contribution strenght. In deformable RoI pooling, thanks to the introduction of learnable bin offsets, a much larger proportion of bins cover the object foreground, leading to gradients being bigger than in background objects and thus having greater influence on prediction.

- **Error-bounded saliency regions**: we can define it as the smallest image region giving the same response as the full image, within a small error bound. In deformable ConvNets, it has been shown that error-bounded saliency regions are not fully focused on the object foreground, which once again suggests that image content outside of the RoI affects the prediction result, leading to bad object detections and predictions.

[4]Spatial support visualization refers to the ability to visualize and understand the impact of convolutional filters on different spatial locations within an image.

3.5.4 Deformable ConvNets v2

By introducing deformable convolution, the network's ability to model geometric transformation is considerably enhanced, in fact, Deformable Convolutional Neural Networks outperform CNNs in their capacity of adaptation to variations due to scale, pose, viewpoint. Neverthless, as described in the previous points, the resultant adapted receptive field often falls into the opposite problem, since it often extends well beyond the region of interest, causing features to be influenced by irrelevant image content.

For this reason, in 2018, the second version of Deformable ConvNets has been presented, that introduce a learnable modulation mechanism to further strengthening the manipulation of spatial support regions. With it, Deformable ConvNets modulate the convolutional weights based on local context information: the modulation helps the network dynamically adjust the importance of different spatial locations during the convolution operation. In the extreme case, a module can decide not to perceive signals from a particular location/bin by setting its feature weight (also called *amplitude*) to zero. Consequently, image content from the corresponding spatial location will have considerably reduced or no impact on the module output [61], enabling more efficient and context-specific feature extraction.

The modulated deformable convolution can then be expressed as:

$$y(p) = \sum_{k=1}^{K} w_k \cdot x(p + p_k + \Delta p_k) \cdot \Delta m_k \tag{3.7}$$

As we can see, the equation is pretty much the same as the normal deformable convolution one, with the addition of the Δm_k term. In this equation K refers to the number of sampling locations of the convolutional kernel and Δm_k to the learnable modulation scalar for the k-th location. Δm_k lies in the range of [0,1]. Both Δp_k and Δm_k are obtained via a separate convolution layer applied over the same input feature maps x.

The design of modulated deformable RoI pooling is similar:

$$y(k) = \sum_{j=1}^{n_k} x(p_{kj} + \Delta p_k) \cdot \Delta m_k / n_k \tag{3.8}$$

where p_{kj} is the sampling location for the j-th grid cell in the k-th bin, and nk denotes the number of sampled grid cells.

With these improvements, modulated deformable convolutional neural networks are reportedly able to outperform deformable ConvNets, focusing on pertinent image

regions. Here are two key advantages:

1. Adaptive Spatial Support: the ability to capture complex patterns and relationships in the input data is improved thanks to flexible spatial locations; this enables the model to better handle spatial deformations and intricate spatial relationships within images. This is particularly beneficial for tasks such as object detection and semantic segmentation where understanding fine-grained details is essential.

2. Enriched Deformation Modeling and Contextual Information: the modulation term helps assigning importance to the areas of the image which are truly meaningful and influential for the feature extraction phase. This leads to strong contextual understanding, because the model is more robust to variations and its performance on challenging scenes is augmented.

Chapter 4

Proposed Model and Analysis

The following chapter accurately describes the architecture of the implemented model, as well as all the steps that lead to the final results, keeping in mind that given a source domain X, Z (of underwater images perturbed by random noise vectors) and desired domain Y (of enhanced images), the goal was to learn a mapping $G : \{X, Z\} \rightarrow Y$ in order to perform automatic image enhancement and restoration.

4.1 System Architecture

The overall structure of the model is shown in Figure **4.1**. The proposed model is a conditional GAN in which the generator tries to learn by evolving with an adversarial discriminator through an iterative min-max strategy. The network takes the name of *DeformUGAN*, because of the structure that it adopts ("U" stands for "underwater") and both the generator and discriminator use modules of the form $convolution - BatchNorm - activation$.

4.1.1 Generator

The generator is a fully convolutional neural network that can resemble the structure of a U-Net in some ways, since it's made of an encoder which downsamples the input image, and a decoder that processes the feature maps performing upsampling and generating the output image having the same dimension of the input (in the case of the project, training pictures had dimension of *256x256 pixels*). Nevertheless, this encoder-decoder structure has some differences with respect to a simple U-Net. In fact, being the U-Net architecture invented for semantic segmentation tasks in biomedical image processing (specifically for the segmentation of neuronal structures in electron microscopic scans of brain tissue), it could lead to important detail and

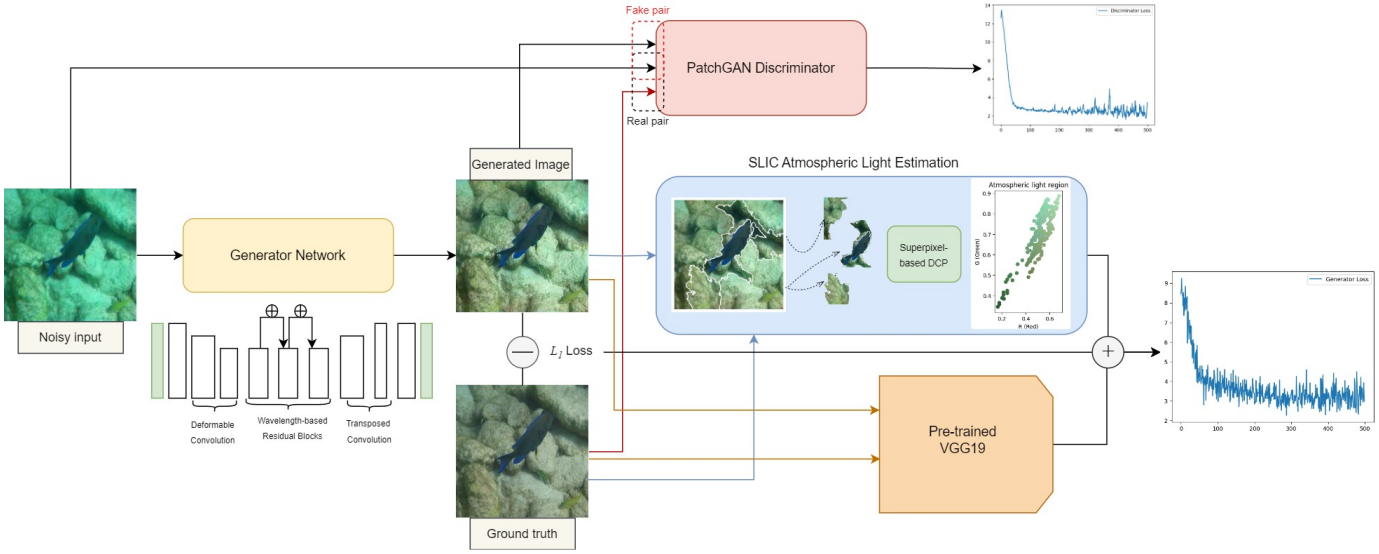


Figure 4.1. Overall Model Architecture.

information loss in underwater images, which are semantically more complex. This is why some adjustments have been added in order to consider all the important details and needs of undersea image processing. More specifically:

- Given a kxk source perturbed underwater image, the generator must learn to produce a kxk output, which has the same content as the input image, but it's perceptually clearer than the input and it's similar to a ground truth image of the latter size;
- The encoder network is made of downsampling stages in the form of *deformable convolution*;
- The decoder network performs upsampling through strided *transposed convolution*, which doubles the spatial dimensions twice;
- Encoder and decoder are linked by a bottleneck structure made of *residual connections*, in order to effectively preserve high-level information across deep layers and to maintain the structure of the input image.
- Both the encoder and decoder pad image with *reflection padding*. This ensures, at the beginning, that the convolution operation doesn't result in the loss of important edge information by reducing the size of the image too quickly, and in the end that the output image has the same spatial dimension of the input one;

Especially, the generator takes as an input an image resized to 3x256x256, converted to a tensor type and whose values are normalized in the range [-1,1] to get data within a range and reduce the skewness which helps learn faster and better.
The architecture of the generator firstly consists of three convolutional layers, two of which use Deformable Convolutions-v2 that reduce the spatial dimensions of the input. This approach enables the encoder to learn various levels of information and produces a more stable and flexible output. In particular, deformable convolution has been adopted in many on-ground images dehazing and denoising works ([53], [57]) but very little research is available on its usage in underwater image enhancement. In the project, it was adopted in the downsampling stages for several reasons: first of all, deformable convolution can capture more abstract representations, since the kernel is dynamic and flexible. This means that, when performing downsampling, this particularly helps in capturing the global context and understanding the overall structure of the input data without loosing local information that would be certainly lost in normal and fixed convolution, because elements are hidden by scatter and backscatter effects underwater. Secondly, deformable convolution allows the network to better capture the variations in transparency, color, and texture that happen underwater, leading to improved enhancement results and enriching the representation of complex spatial structures. Then, the output of the third layer is given as input to a series of three wavelength-based dilated Residual Blocks. Such wavelength dependency is better explained in Section **4.1.2**. From this point onward, upsampling is performed using transposed convolutional layers. Padding layers help maintain the spatial information and symmetry of the input and output images throughout the generator network. Finally, the multi-channel result of this process is combined back into three image channels.

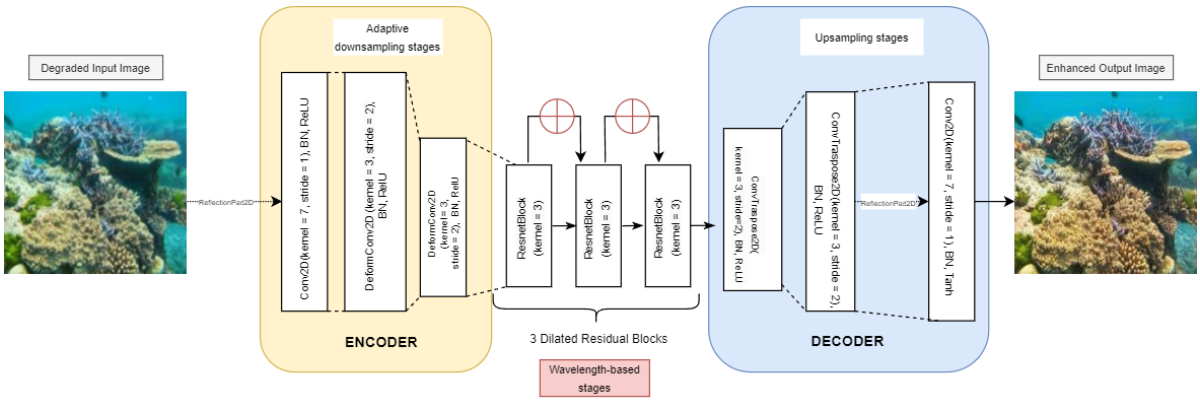


Figure 4.2. Generator Architecture.

4.1.2 Wavelength-dependency

The generator contains a chain of three residual blocks that process the lower-dimensional data extracted by the encoder. The residual blocks have been added in order to make the network learn and keep important large-scale structures within the images; in fact, thanks to them, edges and contrasts are definitely more visible in the output than what would happen in a U-Net structure. Together with this, another important aspect is that such residuals supervise the color channels of the degraded input image with different contextual sizes, considering its local and global semantics based on its *non-uniform* attenuation range. In fact, this non-uniformity of light depends on the wavelength of colors traversing through water and is the reason why underwater images are dominated by blue color (due to its shorter wavelength with respect to green or red, as explained in detail in Chapter **1**).
It has been widely established that the size of the receptive field plays a vital role in high-level vision tasks, especially classification and segmentation that include dense per-pixel predictions, but also in image enhancement, and most of the best-published underwater image enhancement and restoration works process color channels of the degraded images with equal receptive field sizes (alias context). However, similar receptive fields for different color channels may not be a beneficial setting, typically for underwater scenarios.

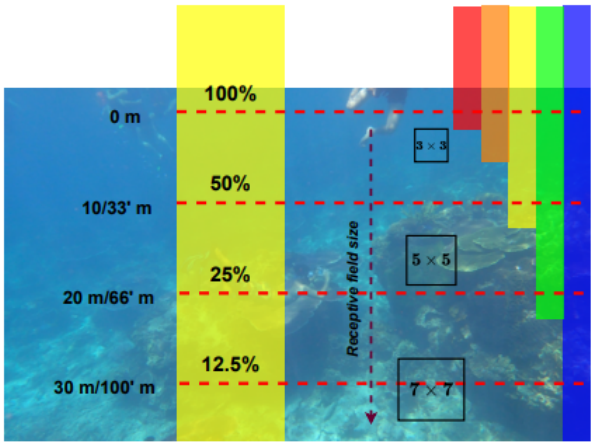


Figure 4.3. Wavelength and Receptive field size. Graphic demonstration of attenuation rates corresponding to different wavelengths of light as it propagates through the water. The blue color traverses the longest because of its shortest wavelength. It is one of the main reasons why underwater images are prevailed by the blue color (Image from: [39])

In fact, the work [39] showed how the attenuation range of underwater colors can be captured through their relationship with the receptive field and how varying its size can be beneficial for the acquisition of all the channel-related information. As it's shown in Figure **4.3** this means that the larger the chosen kernel size, the higher the attenuation range that is being considered within the image. Red and orange colors are captured by very small receptive field sizes (3x3), greenish and yellowish colors by a medium receptive field size (5x5) and bluish colors by large sizes (7x7). From this principle, a chain of three residual blocks has been implemented: after having extracted intermediate features through the adaptive deformable layers, the aim is to keep the relevant channel features, especially the ones having higher wavelength, which are commonly not visible in underwater images. This is done starting from a residual block having 3-sized kernel convolutions, then passing through a residual block having convolutions with a kernel size of 5 and finally via convolutions having kernel size of 7. For computational purposes, the receptive field size is transformed through dilated convolutions. Dilated convolution is a technique that expands the kernel size by inserting holes between its consecutive elements. It involves pixel skipping, so as to cover a larger area of the input, enabling the network to have a larger receptive field without increasing the number of parameters. The dilation rate determines the size of the gaps, and it is a hyperparameter that can be adjusted. When the dilation rate is 1, the dilated convolution reduces to a regular convolution.
In the network, this is translated in residual blocks having two convolutional layers keeping kernel size of 3, but having a different dilation size of 1, 2, and 3, for each different block.
This benchmark can be expressed by the following formula:

$$y_i = x_{i-1} + \sum_{i=1}^{3} \text{ResBlock}_i(x_i), \quad dilation_{Conv2D \in ResBlock_i} = i$$

where y_i is the final output after the chain of residuals, x_{i-1} is the previous input (output of the previous residual block or the very first input of the chain, that is the output of the ReLU activation function in the second downsampling block) and ResBlock_i is made of two dilated convolutions having $dilation = i$, preceeded by two padding layers to preserve spatial dimensions and succeeded by batch normalization layers and a ReLU activation function (for the first convolution). Additionally, a Dropout layer is adopted in the middle to provide additional regularization within the convolutional block. By randomly dropping out some of the activations, dropout helps to prevent the model from relying too heavily on specific features or neurons,

thus encouraging the network to learn more robust and generalizable representations.

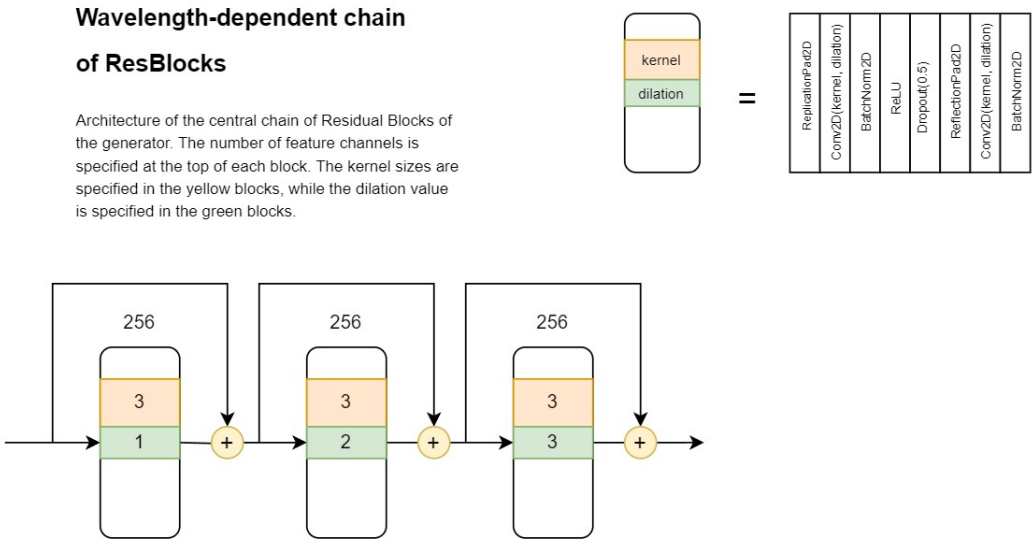


Figure 4.4. Visual representation of the Residual Blocks of the generator.

4.1.3 Discriminator

The network discriminator is a 4-layer Markovian PatchGAN with patches of size 16x16 as desribed in [18]. It's called Markovian because the discriminator considers each patch in isolation, without considering the context provided by neighboring patches. This decreases network complexity and contextually increases the ability of the network to capture high-frequency features such as local texture and style.
Four convolutional layers are used to transform a $256x256x6$ input, that is the size of both the real and generated image expressed in batches, to a $16x16x1$ output that represents the average validity of responses of the discriminator.
This fully-convolutional network halves the spatial resolution of the input four times, while doubling the number of channels. At each layer, 3×3 convolutional filters are used with a stride of 2; then the non-linearity (LeakyReLU) and batch normalization are applied the same way as the generator.

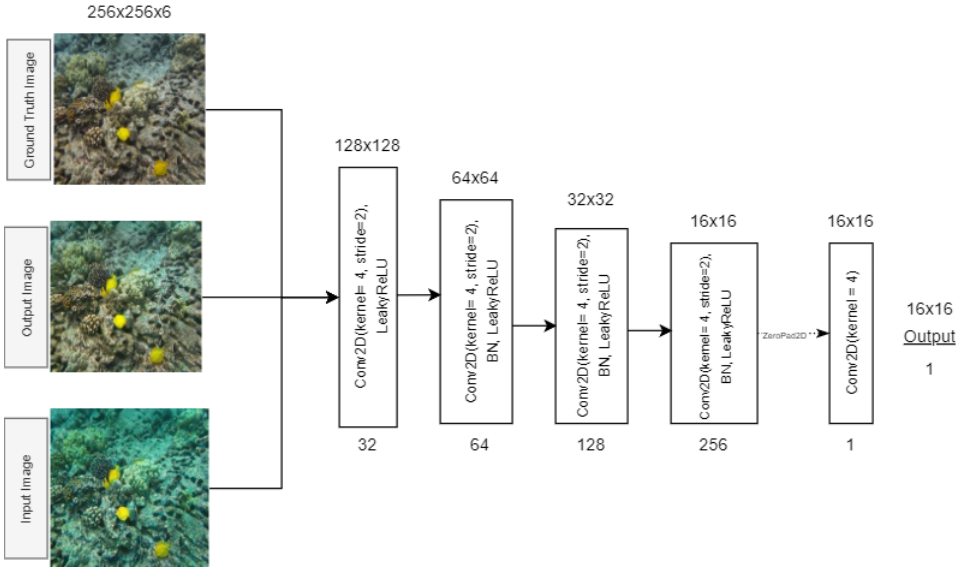


Figure 4.5. Network PatchGAN discriminator.

4.2 Objective Function Formulation

As previously mentioned, DeformUGAN is trained in a min-max framework. The generator G learns how to improve the perceptual quality of the input image so that it is close to the respective ground truth in terms of its global appearance and high-level feature representation. Concurrently, the discriminator D tries to maximize its capacity to discern between real images and reconstructions made by G, enforcing the local texture and style consistency thanks to PatchGAN properties. In DeformUGAN, this is the objective function:

$$\mathcal{L}_{DeformUGAN} = \min_{G} \max_{D} \mathcal{L}_{cGAN}(G, D) + \lambda * \mathcal{L}_1(G) + \gamma * \mathcal{L}_{VGG}(G) + \delta * \mathcal{L}_{DCP}(G)$$

Where λ, γ and δ are the weights associated with each loss, tuned as hyperparameters and of values 0.6, 0.2 and 0.2 respectively.
The first term refers to the conditional adversarial loss function, expressed as:

$$\mathcal{L}_{cGAN}(G, D) = \min_{G} \max_{D} [E_{x \sim p_{\text{data}}(x), y \sim p(y)}[\log D(x, y)] + E_{z \sim p(z), y \sim p(y)}[\log(1 - D(G(z, y), y))]$$

where $p_{data}(x)$ is the distribution of the real data samples X from the dataset distorted with a random noise of the distribution $p(z)$ and $p(y)$ is the distribution of enhanced images Y from the dataset (ground truth). Here, the generator G tries to minimize $\mathcal{L}_{cGAN}$ while the discriminator D tries to maximize it.

After the general $\mathcal{L}_{cGAN}$, three other losses have been implemented in order to adjust the problem with the specific objective:

- **Global similarity loss**: the global similarity loss, or $\mathcal{L}_1$ loss creates a criterion that measures the mean absolute error (MAE), or L1 distance, between each element in the input x and target y[1]. In the model, this loss is used between the generated image and the ground truth image, to enhance their pixel-wise similarity. $\mathcal{L}_1$ is expressed as:

$$\mathcal{L}_1(G) = \frac{1}{N}\sum_{i=1}^{N} |Y - G(X,Z)|, \quad N = |pixels \in Y|$$

- **Feature and content loss**: the content loss $\mathcal{L}_{VGG}$ is implemented in order to encourage G to generate enhanced image that has similar content (i.e., feature representation) as the target image. As in [18], the function ϕ tries to minimize the difference between the high-level features extracted by the `block5_conv2` layer of a pre-trained VGG-19 network within G(X,Z) and Y:

$$\mathcal{L}_{VGG}(G) = ||\phi(Y) - \phi(G(X,Z))||_2$$

- **Atmospheric Light loss**: the atmospheric light loss, or DCP loss $\mathcal{L}_{DCP}$, creates a criterion that measures the atmospheric light α based on the dark channel prior of the target image Y and the generated image G(X, Z), expressed as a n-dimensional tensor which contains its RGB values, and minimizes the Euclidean distance, or L2 distance, between the two:

$$\mathcal{L}_{DCP}(G) = ||\alpha(Y) - \alpha(G(X,Z))||_2$$

  Such loss is computed using the SLIC algorithm for superpixels computation.

4.2.1 SLIC (Simple Linear Iterative Clustering) Algorithm

The SLIC algorithm [1] adapts a k-means clustering approach to efficiently generate superpixels. Superpixels are defined as an atomic region of pixels sharing similar characteristics (like intensity). In fact, they are formed by perceptually similar pixels to create visually meaningful entities while heavily reducing the number of primitives for subsequent processing steps [43].

[1] Definition from the Pytorch documentation: L1Loss, `https://pytorch.org/docs/stable/generated/torch.nn.L1Loss.html`

(a) SLIC pseudocode.

Algorithm 1 SLIC superpixel segmentation

/ Initialization */*
Initialize cluster centers $C_k = [l_k, a_k, b_k, x_k, y_k]^T$ by sampling pixels at regular grid steps S.
Move cluster centers to the lowest gradient position in a 3×3 neighborhood.
Set label $l(i) = -1$ for each pixel i.
Set distance $d(i) = \infty$ for each pixel i.

repeat
  / Assignment */*
  for each cluster center C_k **do**
    for each pixel i in a $2S \times 2S$ region around C_k **do**
      Compute the distance D between C_k and i.
      if $D < d(i)$ **then**
        set $d(i) = D$
        set $l(i) = k$
      end if
    end for
  end for
  / Update */*
  Compute new cluster centers.
  Compute residual error E.
until $E \leq$ threshold

(b) SLIC with k = 10 on training samples.

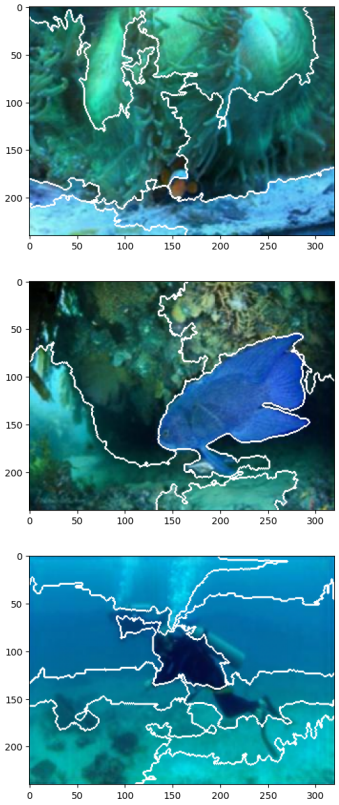


Figure 4.6. SLIC and examples of its applications within the project images.

Superpixels are becoming popular in computer vision applications because:

- They carry more information than pixels; in fact, pixels alone aren't very informative and don't hold any sort of semantic meaning;
- Superpixels are meaningful units because they group together pixels that exhibit comparable visual characteristics, enhancing our perception of coherent regions within an image;
- They provide a convenient and compact representation of images that can be very useful for computationally demanding problems (such as image segmentation or object detection).

In particular, the SLIC algorithm clusters pixels in the combined five-dimensional color and image plane space [labxy] (in which [lab] is the pixel color vector in CIELAB color space and xy is the pixel position) to efficiently generate compact,

nearly uniform superpixels. The CIELAB color space, also referred to as $L*a*b*$, is a color space that expresses color as three values: $L*$ for perceptual lightness and $a*$ and $b*$ for the four unique colors of human vision: red, green, blue and yellow. So, a superpixel is a single value C_k equal to $[L_k^*, a_k^*, b_k^*, x_k, y_k]$, where k is a value between 1 and the total number of desired superpixels K. Another parameter that can be defined is the *compactness*: it determines the extent to which the superpixels adhere to their initial shape, affecting their regularity. A lower compactness value allows the superpixels to better adhere to image boundaries, resulting in irregular shapes. Conversely, a higher compactness value encourages superpixels to be more compact and regular, potentially leading to smoother boundaries and larger superpixel sizes. SLIC superpixel segmentation is described in Figure **4.6a**: it initially sets K cluster centers in a NxN dimensional image and navigates through each cluster center C_k in an iterative loop that repeats until stability. Then, it finds similar pixels in a fixed-size neighborhood (2Sx2S) and computes new centers, where $S = \sqrt{N/K}$ is the distance between pixel centers. When stability is achieved, superpixels are defined. The SLIC algorithm finds application in various fields such as image segmentation, object tracking, but can also be efficiently used in tasks in which it's necessary to consider relevant image regions, as in dark channel prior computation for atmospheric light and transmission estimation.

4.2.2 Atmospheric Light Estimation

$\mathcal{L}_{DCP}$ has the aim of minimizing the Euclidean distance (L_2 distance) between the atmospheric light estimated by the ground truth image and by the generated one. In order to implement atmospheric light, a superpixel-based dark channel computation method is proposed. The method is inspired by the works [55] and [15], that however uses it in on-ground hazy images; for this reason, it's been necessary to adapt it to underwater environments.

In the project, SLIC is computed using 10 cluster centers and compactness = 5 and is firstly used to identify image superpixels. They will be necessary for the computation of the dark channel and consequently of the atmospheric light. In conventional studies, most dehazing methods use the brightest pixel in a single hazy image to represent atmospheric light, which neglects saturated pixels, especially pixels in the white object regions. To minimize the influence of the above issue on atmospheric light estimation, [15] estimated atmospheric light by picking the top 0.1 percent of the brightest pixels in the dark channel. But the method uses local patches, and this could still bring mistakes. This is why SLIC superpixels are

```python
def superpixel_atmospheric_light(segment_pixels, img):
  min_superpixels = []
  #estimating the dark channel prior for each superpixel area
  for i in segment_pixels.keys():
      #we pick the smallest value for each channel and
      #each superpixel --> Dark Channel Prior
      min_superpixels.append(np.min(np.array(segment_pixels[i]), axis = 0))
  #numpy flattened array containing all the minimum pixels
  flatdark = np.array(min_superpixels).ravel()
  #we sort it to take the maximum elements afterwards
  flatdark.sort()
  #0.3 percentage of pixels estimating the atmosphere light
  if flatdark.shape[0] > 3:
    p = int(0.3*(flatdark.shape[0]))
  else:
    p = flatdark.shape[0]
  # find top M * N * p indexes
  searchidx = (-flatdark).argsort()[:p]
  #print('atmosphere light region:',
  # np.max(img.numpy().take(searchidx, axis = 0), axis = 0))
  # return the highest intensity for each channel
  flat_img = img.reshape(256 * 256, 3)
  return np.max(flat_img.numpy().take(searchidx, axis = 0), axis = 0)
```

Figure 4.7. SLIC-based atmospheric light computation in the project.

used instead of fixed-size local patches: they combine adjacent pixels with similar texture, colour, brightness and other characteristics into one region and every region is used to compute the dark channel. After having performed SLIC superpixels segmentation, the algorithm performs as follows:

1. Dark channel is estimated returning the minimum value for every color channel in every superpixel;

2. All the RGB values are put in a single numpy array containing all the minimum pixels and the flattened array is sorted in ascending order;

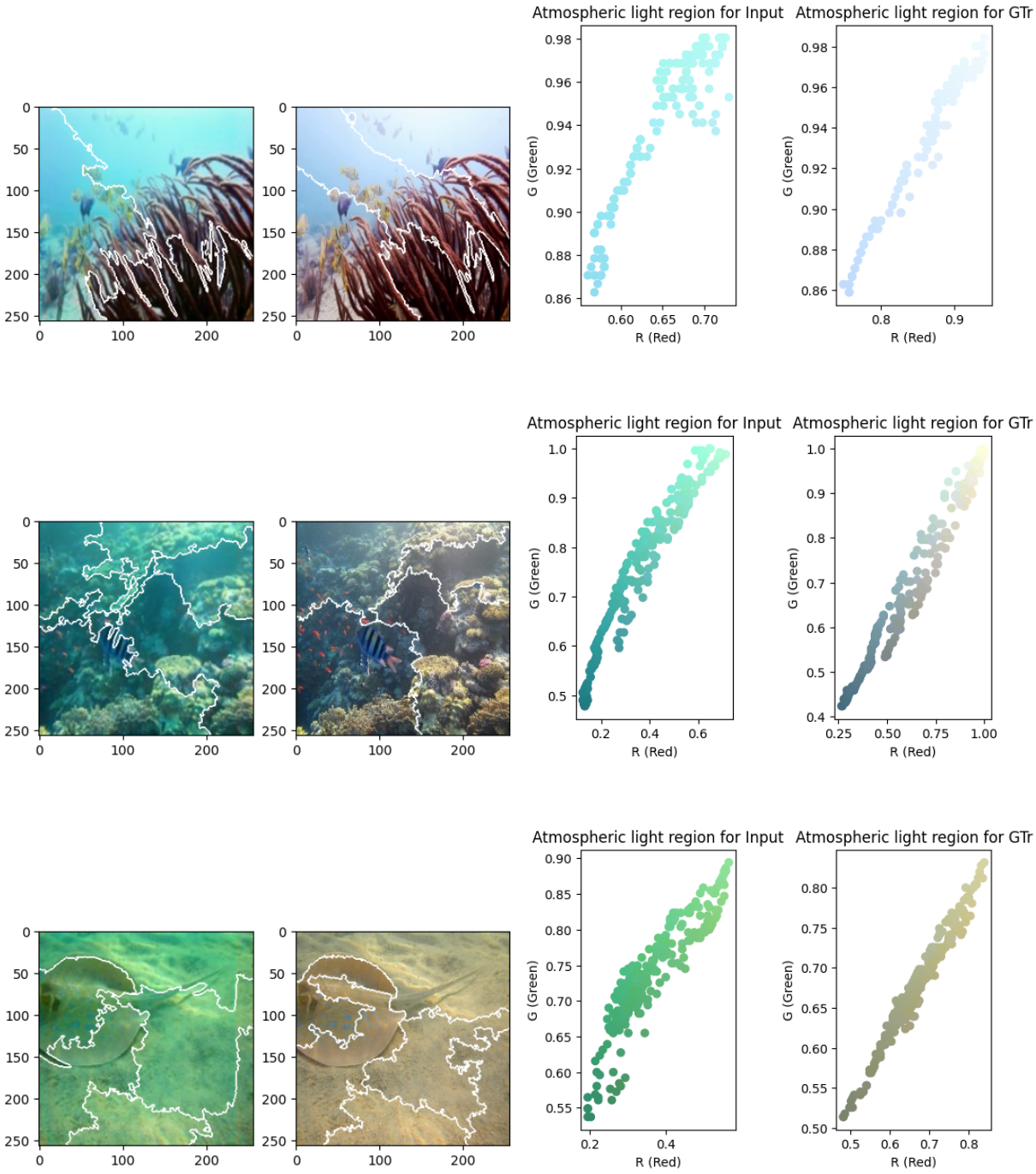
3. The 0.3 percent of the elements of the array is calculated in order to eventually take the 0.3 percent of brightest pixels. In fact, this index is not fixed since it strictly depends on the number of superpixels found within the image. For this reason, a particular control on the length of the numpy array is done to prevent the algorithm considering just one pixel in cases where 3 or less superpixels are found;

4. The top M * N * p indexes are found, which correspond to the top N indexes of the numpy array that describe the atmospheric light, where $N = 0.3$ percent of the length of the array previously computed;

5. The atmospheric light region is estimated through all the indexes;

6. Atmosphere light is returned taking the largest value for each channel in the region;

Some examples of the results of atmospheric light estimation in the project are shown below in a comparison between the input noisy image and the ground truth.



It's visible how SLIC algorithm performs differently in such images, enhancing different areas and finding different edges and regions. In addition, as it can be noticed from the plot, the atmospheric light regions detected by the algorithm are quite different from each other, thus representing a valuable source of information to be considered in the optimization phase for restoration.

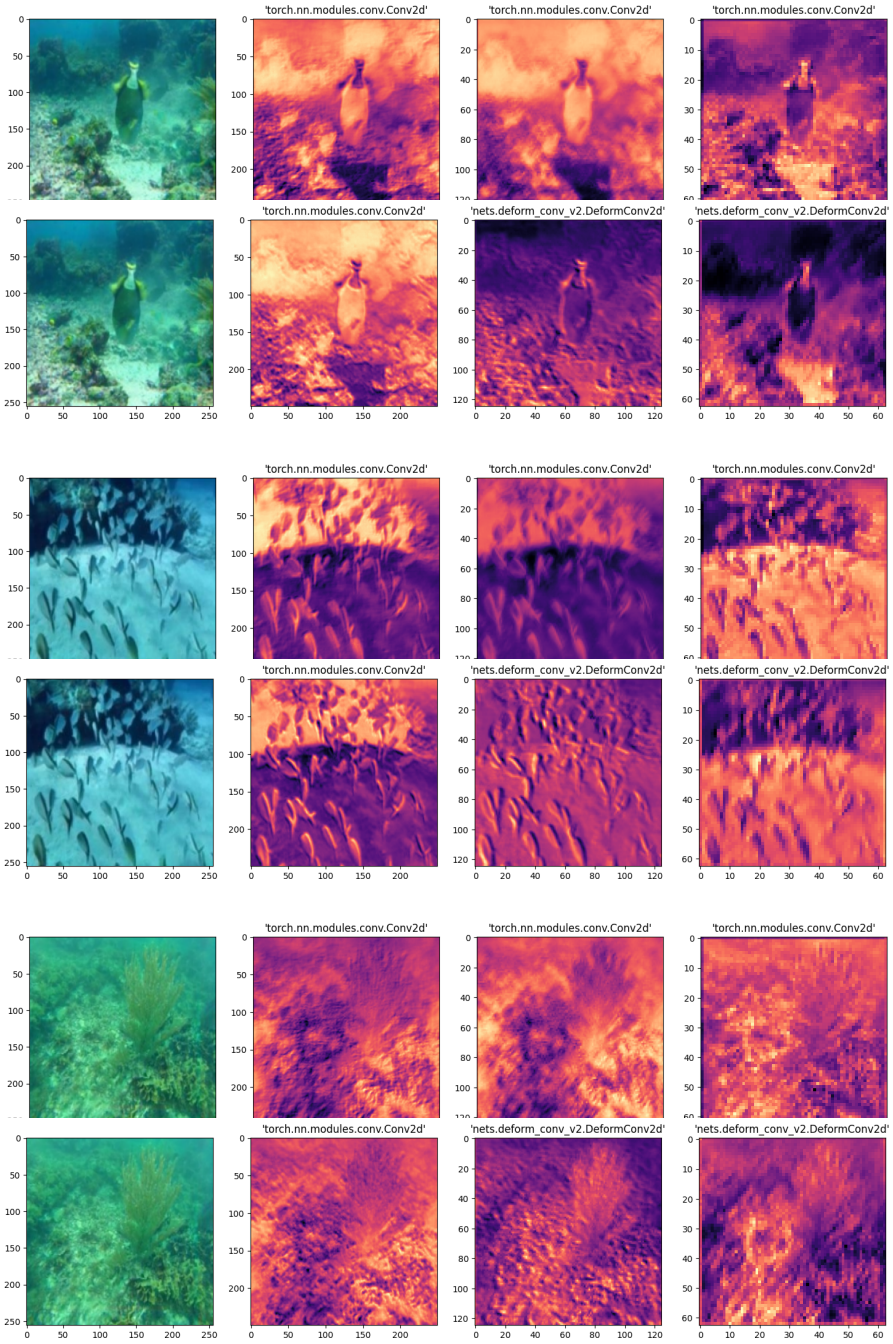
Chapter 5

Experimental Results

5.1 Ablation Study

Before finding the ultimate and definitive model, some experiments have been carried out in order to define the correct hyperparameters and the most suitable architecture structure. All these experiments have been conducted training the model with 3000 instances of the EUVP dataset (which were found to be enough for such a phase) for 100 epochs with a GPU NVIDIA Tesla T4 for AI inference. The ablation study was conducted with Google Colaboratory, a cloud-based platform provided by Google that offers free access to a Jupyter notebook environment along with free GPU and TPU support[1] and using the PyTorch framework[2].
Firstly, it was examined the impact of deformable layers, dilation, dropout and $\mathcal{L}_{DCP}$ on the model. The influence of deformable layers has been investigated by plotting the activation maps resulting from the encoding part having all convolutional layers and then replacing the two convolutional layers with two deformable convolutional layers when performing downsampling. More precisely, the visualizations depict the feature maps generated by the different convolutional layers, specifically focusing on the 64th, 128th, and 256th layers (the last ones of each layer), respectively, across five samples extracted from the *EUVP* dataset. The images were chosen over the ones which capture really noisy environments, in order to show how such layers behave on complex data.

[1] Google Colaboratory, `https://colab.research.google.com/`
[2] PyTorch, `https://pytorch.org/`



As can be noticed in the plots, the first deformable layer clearly captures edges and corners in images, effectively preserving local contrast information, which is important for the understanding of the spatial layout of objects and sampling locations. On the other hand, the convolutional layer seems to capture more general features and suffers from the degradation of underwater images, failing to capture any structural information in the image.

The second deformable layer, having outlined important visual cues before, enhances contrasts. This also happens in the respective convolutional layer, but the deformable one evidently amplifies them by making the foreground regions darker. This leads to think that deformable convolution facilitates discrimination between foreground and background, in order to better distinguish elements within the global image context.

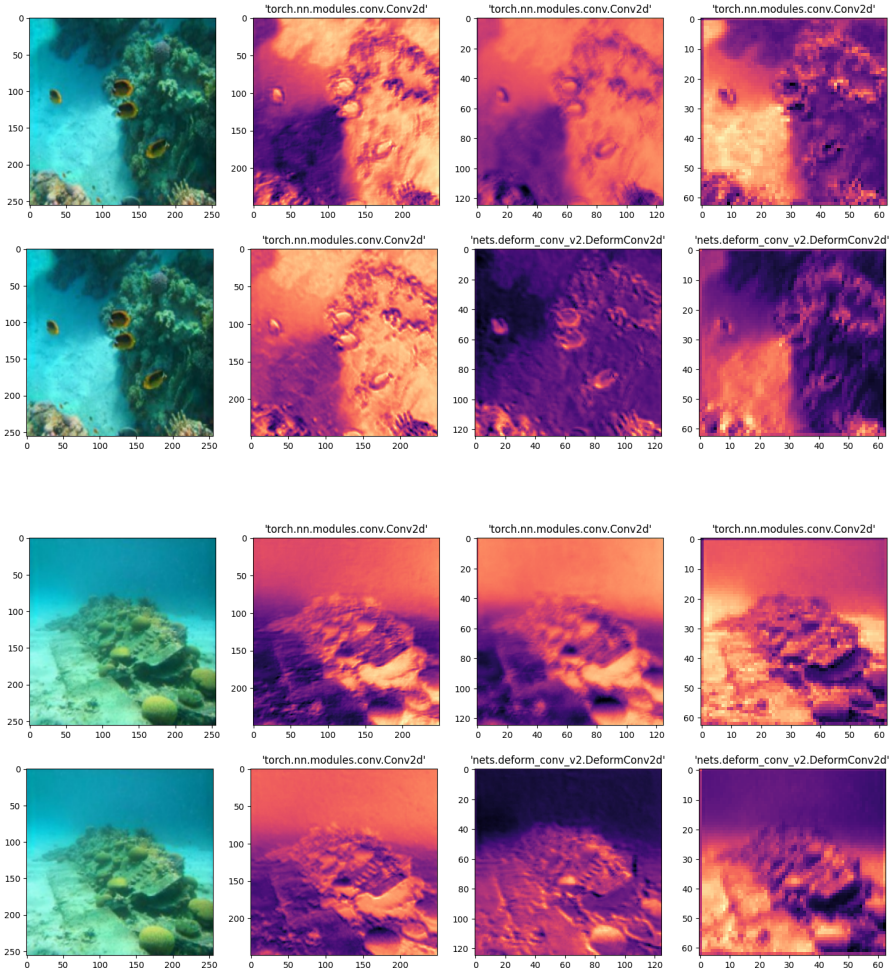


Figure 5.1. Ablation experiment on the feature maps of the downsampling layers: the first picture of every pair shows downsampling with convolutional layers and the second one shows downsampling with deformable layers.

Then, the impact of adaptive and wavelength-based residuals with dropout has been studied. The residual chain has the goal of capturing and keeping colors' information based on their different wavelengths traversing through water, through adaptive kernel dimensions. For this reason, the outputs of DeformUGAN with this adaptiveness have been compared with the same model having non-adaptive

residuals (all the layers of the chain have a receptive field of dimension 3x3).

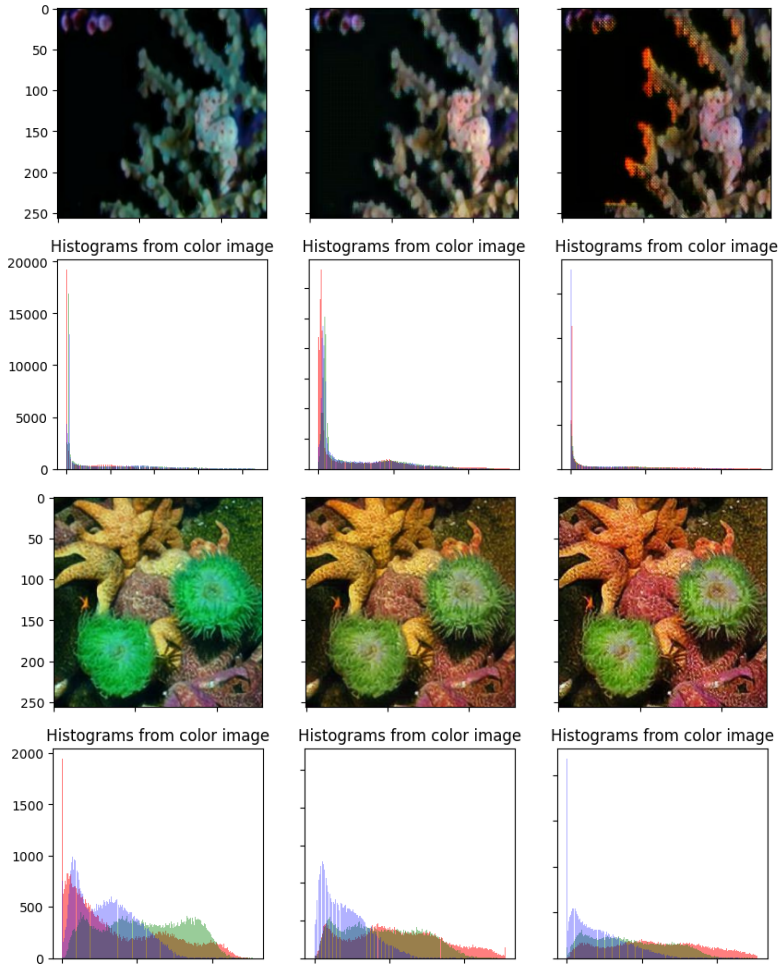
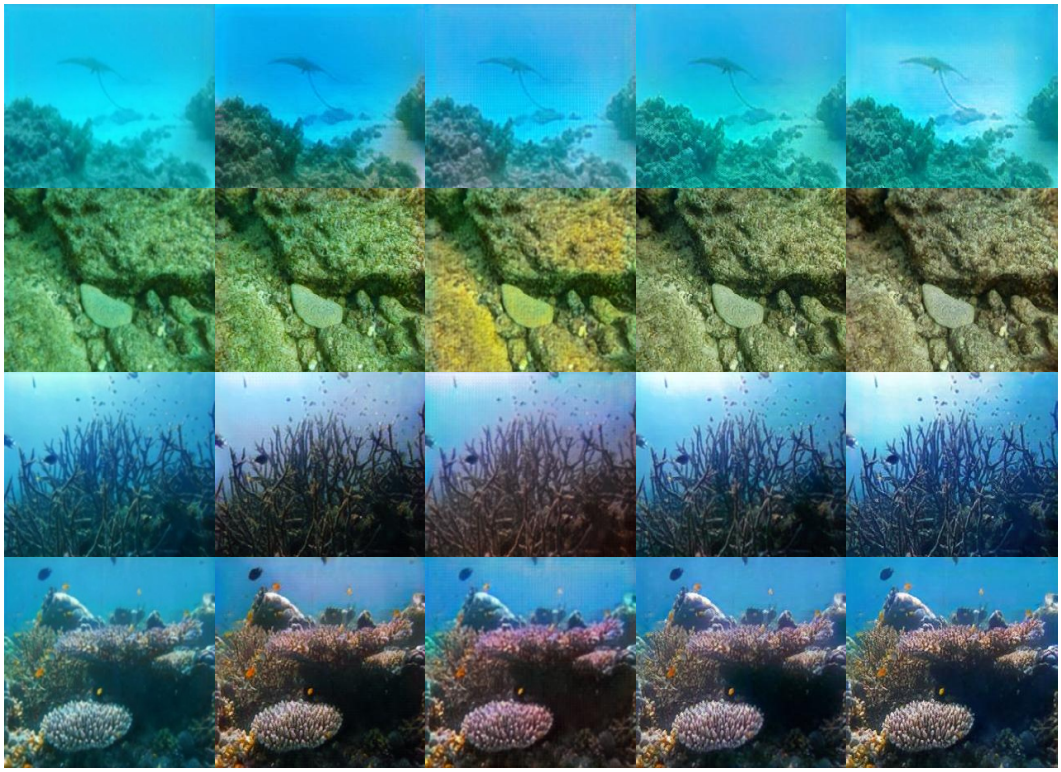


Figure 5.2. Ablation experiment on the adaptiveness of residuals. From left to right: noisy underwater image, restored image with non-adaptive residuals and restored image with dilated residuals (and their corresponding RGB histogram distributions).

Images and plots show the testing results of each model at the 100th epoch. As can be noticed, dilation helps enhance saturation values, especially the ones related to the red channel. Finally, the ablation experiment concluded with a deep study of the learning enhancement based on the different impact of the different losses attributes. The model was trained without (w/o) $\mathcal{L}_1$, $\mathcal{L}_{VGG}$ and $\mathcal{L}_{DCP}$, respectively, in order to show the contributions of each loss term for the performance of the network. As it's shown in Figure **5.3**, we can observe that global similarity loss ($\mathcal{L}_1$) helps improve the overall illumination of the image, the content loss ($\mathcal{L}_{VGG}$) helps to generate images with enhanced contrast and accurate color reproduction and atmospheric light loss ($\mathcal{L}_{DCP}$) stabilizes RGB values, sharpness and clarity.



Input | w/o $\mathcal{L}_1$ | w/o $\mathcal{L}_{VGG}$ | w/o $\mathcal{L}_{DCP}$ | DeformUGAN

Figure 5.3. Ablation experiment: learning enhancement without (w/o) $\mathcal{L}_1$, $\mathcal{L}_{VGG}$, and $\mathcal{L}_{DCP}$ loss terms in DeformUGAN

5.1.1 Hyperparameter Tuning

LR Range Test The Learning Rate Range Test (LRRT) is a method for discovering the optimal learning rate values that can be used to train a model without divergence through a pre-training run in which the learning rate is increased linearly or exponentially between two boundaries. This method was used to find the best learning rates for the learning process of both the generator G and the discriminator D.

The training pipeline was executed for 500 iterations, and the initial learning rate was set to $1e^{-3}$ both for G and D. LRRT was performed with the contribution of the learning rate scheduler *LambdaLR*, which allows for a custom learning rate scheduling strategy. In this case, it scales the learning rate linearly with respect to the number of iterations performed. Inside the loop, after each optimization step (i.e., after updating the parameters of the generator and discriminator networks), the respective scheduler is invoked to update the learning rate and, at the end, the

learning rate associated to the minimum loss value is chosen as the one to train the architecture.
The learning rates chosen after the LRR Test were 0.000287 for the generator and 0.000425 for the discriminator.

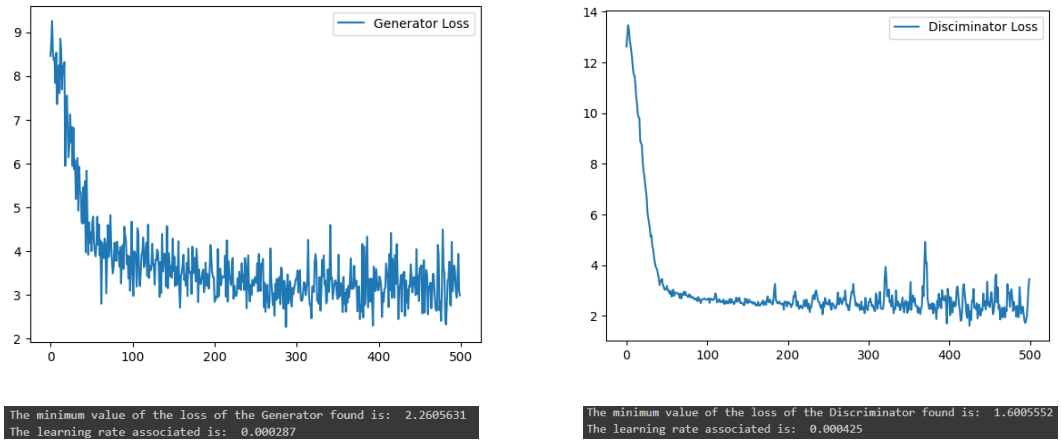


Figure 5.4. LLRT execution over 1000 iterations for DeformUGAN generator and discriminator.

Epochs, batch size and superpixel cluster centers Subsequently, a further analysis on the number of epochs of training, batch size and number of superpixel cluster centers k has been conducted over the whole dataset. Below is the list of all the tested configurations:

- Training for 100 epochs, batch size = 6, k = 10;
- Training for 70 epochs, batch size = 10, k = 10;
- Training for 70 epochs, batch size = 6, k = 12;
- Training for 70 epochs, batch size = 6, k = 15;

Where k is the number of cluster centers detected by the SLIC algorithm. *Adam* optimizer with $\beta_{1,2}$ = (0.5, 0.99) was used in all these configurations, and 100 epochs coincide with nearly 70k iterations. All these training sessions weren't conducted using Google Colab due to its computational limitations. Instead, they were carried out with a Nvidia RTX 4090 GPU with 24GB of VRAM and an Intel i9 13400 CPU on the whole *EUVP Dataset*.

5.2 Quantitative Evaluations

In order to show the performance of the DeformUGAN model, it was compared with other state-of-the-art algorithms for underwater image enhancement and restoration on the testing set of the EUVP dataset, composed of 515 images of different quality and diversified situations. Firstly, a comparison of the performance of the model with different configurations was performed with the following evaluation metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), Patch-based Contrast Quality Index (PCQI), Underwater Image Quality Measure (UIQM), Underwater Image Sharpness Measure (UISM), Underwater Color Image Quality Evaluation (UCIQE), Average Entropy (E), Natural Image Quality Evaluator (NIQE). Excluding the ones already described in the dedicated section in Chapter **2**, here's a brief description of the other metrics included:

- Average Entropy (E): it measures the randomness of pixel values. Especially in underwater environments, a high E usually indicates degraded images covered by layers of backscatter. The formula is given by:

$$E = -\sum_{i=0}^{L-1} p(i) \log_2(p(i))$$

  where $p(i)$ is the probability of occurrence of intensity level i and L is the number of number of intensity levels (256 for 8-bit images).

- Natural Image Quality Evaluator (NIQE): it calculates the no-reference image quality score for low-quality images, making use of only measurable deviations from statistical regularities observed in natural images, without training on human-rated distorted images. A smaller score indicates better perceptual quality (a NIQE equal to 0 is the best).

The Table **5.1** shows quantitative results of the hyperparameter tuning ("e" and "b" stand for number of epochs and batch size, whereas "k" stands for the number of superpixel cluster centers). The evaluation metrics displayed in cyan are the ones strictly related to the underwater environment, whereas in bold are the best results achieved over the specific metric.
The table shows that increasing the number of cluster centers can effectively improve the performance of the model, even with a lower batch size value and less epochs of training.

Table 5.1. Comparison of the performance of the method over different configurations (↓ means the lower the better).

Method	PSNR	UIQM	PCQI	SSIM	UISM	UCIQE	E ↓	NIQE ↓
DeformUGAN b6 - e80 -k10	26.47	2.936	0.693	0.818	6.866	0.583	7.487	51.54
DeformUGAN b6 - e90 -k10	26.02	2.916	0.694	**0.822**	6.949	0.579	7.481	51.32
DeformUGAN b10 - e50 -k10	26.72	2.911	0.701	0.821	6.764	0.583	7.474	50.41
DeformUGAN b10 - e60 -k10	26.15	2.845	0.713	0.819	6.752	**0.592**	7.508	**49.64**
DeformUGAN b10 - e70 -k10	26.36	2.937	0.707	0.819	6.881	0.590	7.491	51.20
DeformUGAN b6 - e60 -k12	25.83	**2.997**	0.697	0.807	**6.975**	0.588	7.504	52.67
DeformUGAN b6 - e70 -k12	**26.73**	2.846	**0.725**	0.813	6.714	0.591	7.520	52.25
DeformUGAN b6 - e60 -k15	26.46	2.916	0.713	0.821	6.881	0.589	7.489	52.41
DeformUGAN b6 - e70 -k15	26.17	2.949	0.713	**0.822**	6.827	0.583	**7.471**	51.39

The DeformUGAN models that had the highest scores were chosen to be compared with the state-of-the-art models over the same evaluation metrics (Figure **5.2**). Metrics' results were rounded to the nearest hundredth in order to be in line with the other works' results. The names of the task-related evaluation metrics are displayed in cyan. The best and second-best results are shown in **bold** and underlined, respectively. As can be noticed, the model outperforms state-of-the-art models in two evaluation metrics: UCIQE and PCQI. This means that DeformUGAN is particularly able to restore and enhance areas of contrast and RGB values. Especially for PCQI, the model achieves much higher values with respect to other noticeable works used in research. Another visible salient aspect is that all the other metrics' values are really near the state-of-the-art: in fact, for all of them DeformUGAN reaches the second top value or the third, at most.

SSIM and PSNR metrics are computed in Table **5.3**. DeformUGAN isn't able to surpass the state-of-the-art results but, even in this case, it reaches the top second

Table 5.2. Comparison of different SOTA methods on various metrics.

Method	UIQM	UISM	UCIQE	PCQI	E ↓	NIQE ↓
UGAN – SOTA 2018	2.89	6.84	.581	.700	7.52	49.90
UGAN-P	2.93	6.83	.590	.704	7.54	50.17
FUnIE-GAN – SOTA 2020	2.97	6.90	.590	.706	7.55	50.51
FUnIE-GAN UP	2.93	6.86	.588	.702	7.50	52.87
Deep SESR - 2020	**3.09**	**7.06**	.572	.679	7.40	55.68
Deep WaveNet -2022	3.04	**7.06**	.559	.706	**7.38**	**44.89**
DeformUGAN b10 - e60 -k10	2.85	6.75	**.592**	.713	7.51	49.64
DeformUGAN b6 - e60 -k12	3.00	6.98	.588	.697	7.50	52.67
DeformUGAN b6 - e70 -k12	2.85	6.71	.591	**.725**	7.52	52.25
DeformUGAN b6 - e70 -k15	2.95	6.83	.583	.713	7.47	51.39

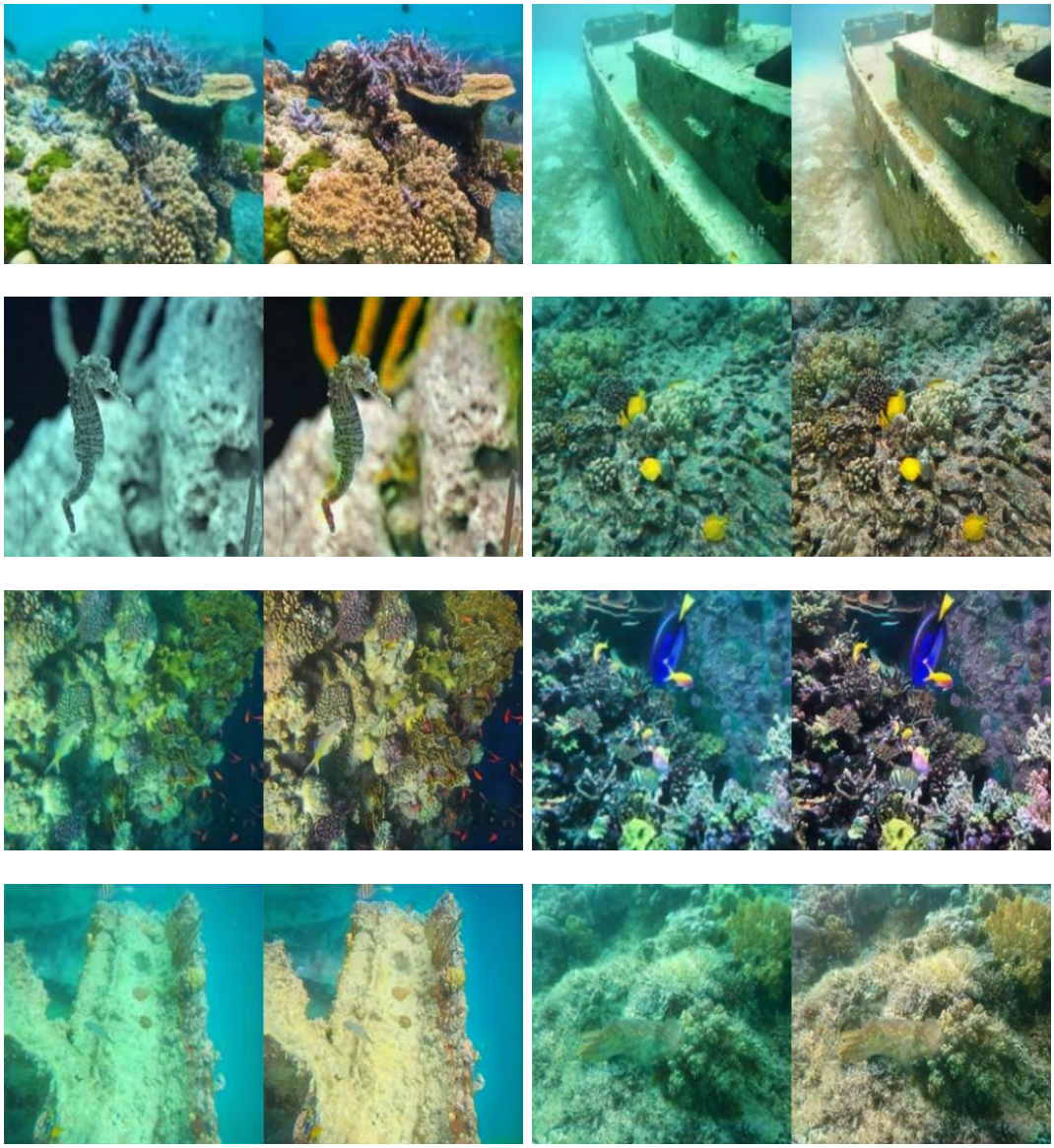
ranking for PSNR and third for SSIM. In particular, for SSIM the margin between its performance and the top-ranked results is merely 0.01. This outcome suggests that despite falling short of the current state-of-the-art for this specific evaluation metric, DeformUGAN demonstrates considerable potential for further advancements in research.

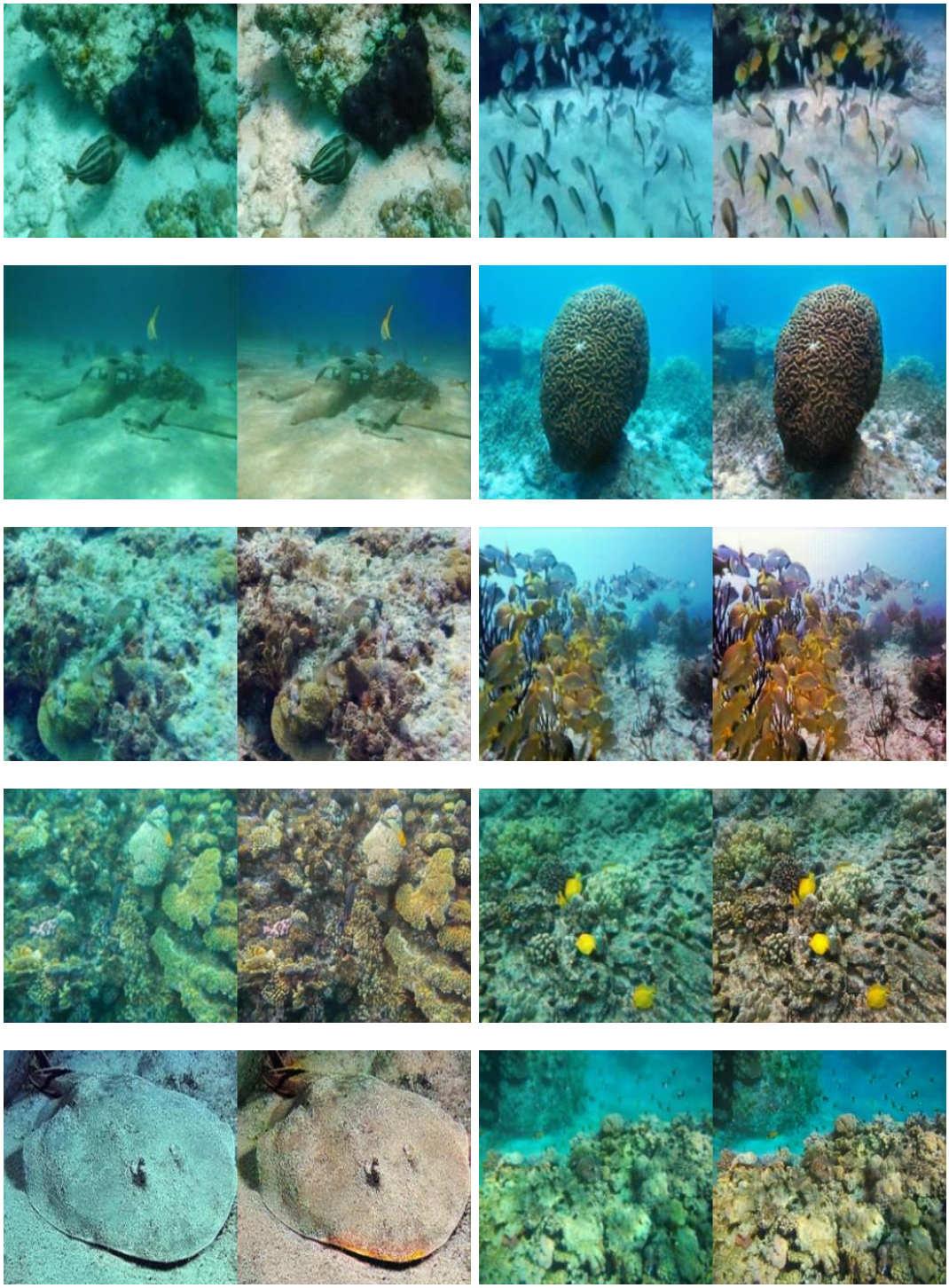
Table 5.3. Comparison of different SOTA methods on SSIM and PSNR metrics.

Method	Dataset	PSNR	SSIM
UGAN – SOTA 2018	EUVP	26.55	.80
WaterNet - 2019	EUVP	25.28	.83
FUnIE-GAN – SOTA 2020	EUVP	26.22	.79
Deep SESR - 2020	EUVP	25.25	.75
Shallow UWNet – SOTA 2021	EUVP-Dark	**27.39**	.83
RAUNE-Net - 2023	EUVP	26.33	**.84**
DeformUGAN b10 - e60 -k10	EUVP	26.15	.82
DeformUGAN b6 - e60 -k12	EUVP	25.83	.81
DeformUGAN b6 - e70 -k12	EUVP	26.73	.81
DeformUGAN b6 - e70 -k15	EUVP	26.17	.82

5.3 Qualitative Evaluations

Below are some interesting results of the implemented model over degraded underwater images. As can be noticed, edges and boundaries are enhanced and most of the colors are restored. Even in pictures with deep bluish predominances, the network smoothly recovers RGB pixel values with considerable high fidelty.
The main enhancements regard the brightness improvement, colorization and detail recovery.





An enhanced image with high brightness, high contrast, and full color can be useful for applications in advanced vision tasks, such as object detection or 2D pose estimation applications. To demonstrate that the enhanced images effectively enrich the edge and feature information, SIFT feature point matching and Canny operator

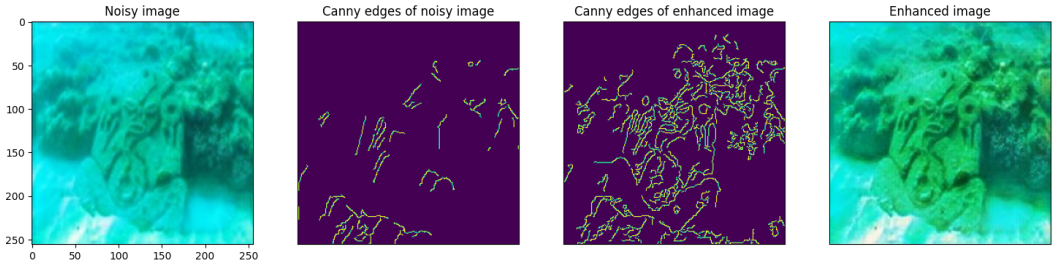
edge extraction were performed on degraded images and enhanced images, along with the plot of RGB distribution, which shows how the variation in RGB color intensity is adjusted with enhancement and restoration.

5.3.1 Canny Edge Extraction

Canny is a popular edge detection algorithm used in feature extraction for many computer vision tasks such as image segmentation, object detection and scene understanding. It involves the below-mentioned steps:

1. Noise and degradation reduction using a 5x5 Gaussian filter;
2. Computation of the derivative of the Gaussian filter to calculate the gradient of image pixels in order to obtain intensity along x and y dimensions;
3. Non-maximum suppression to remove any unwanted pixels which may not constitute the edge. For this, at every pixel, the pixel is checked if it is a local maximum in its neighborhood in the direction of gradient;
4. Hysteresis Thresholding method to preserve the pixels higher than the gradient magnitude and neglect the ones lower than the low threshold value.

In the project, the lower and upper values for hysteresis thresholding are set to 100 and 200, respectively. The plots below show how the algorithm is able to enhance very degraded and backscattered images, detecting a huge quantity of edges and details with respect to noisy pictures. This allows monitoring ROVs and AUVs to detect more objects in the scene, as well as better identifying their shape in order to perform other vision tasks.



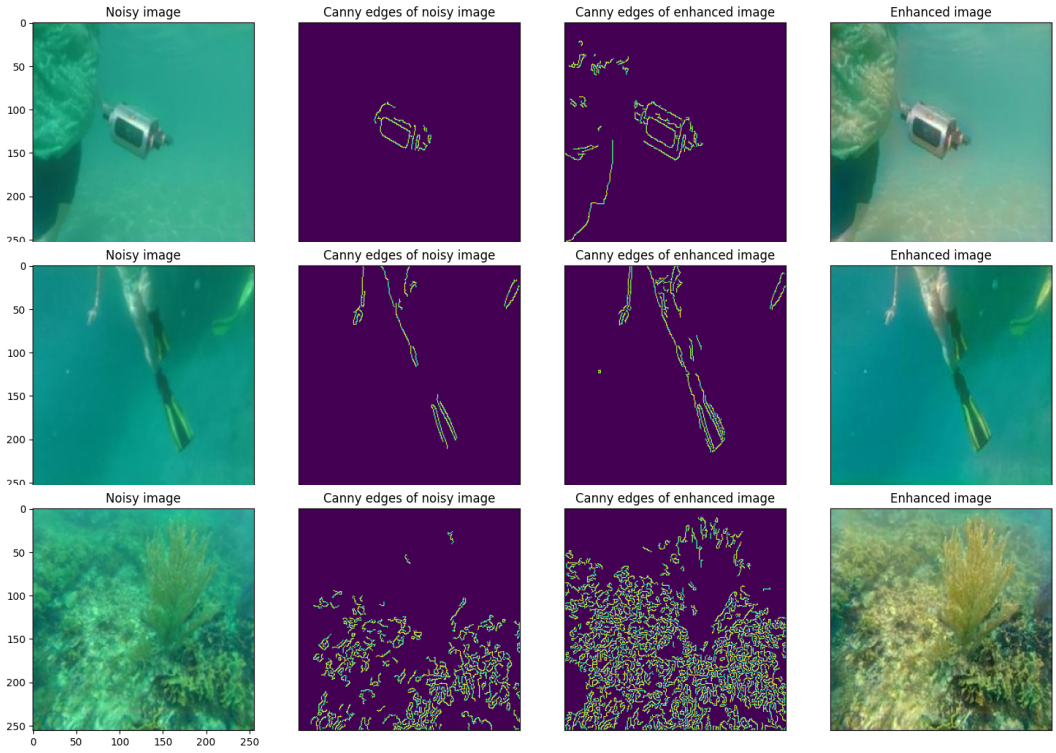
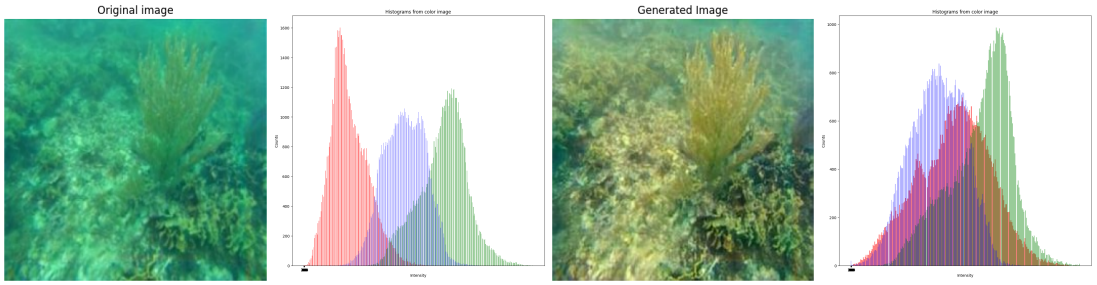


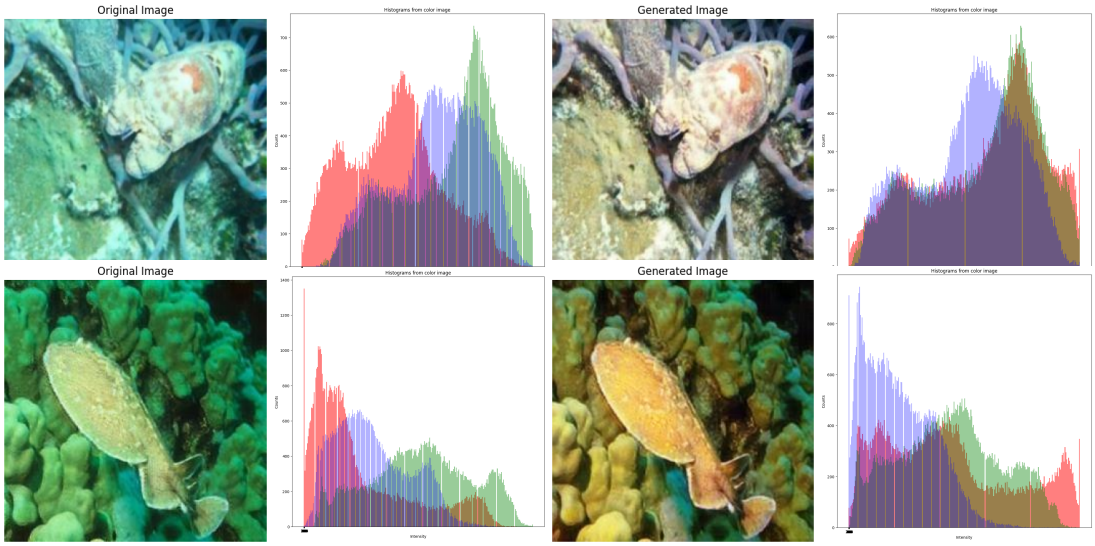
Figure 5.5. Examples of the application of Canny on degraded and enhanced images.

5.3.2 RGB Distribution

The plots of the RGB distribution clearly show how restoration and enhancement are able to repair fine-grained color details, especially balancing red and blue distributions. In fact, red colors appear more visible and recovered. On the other hand, blue and green channels' intensity is diminished.

In the plots, the x-axis represents the intensity of the RGB values for every pixel, expressed in values ranging in [0, 255] and the y-axis represents the quantity of pixels having such intensity.

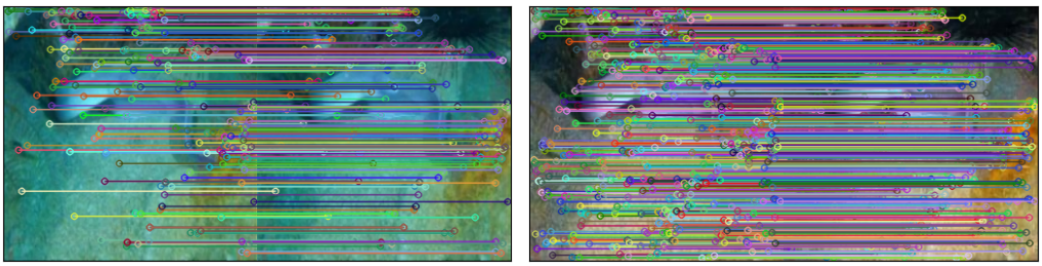




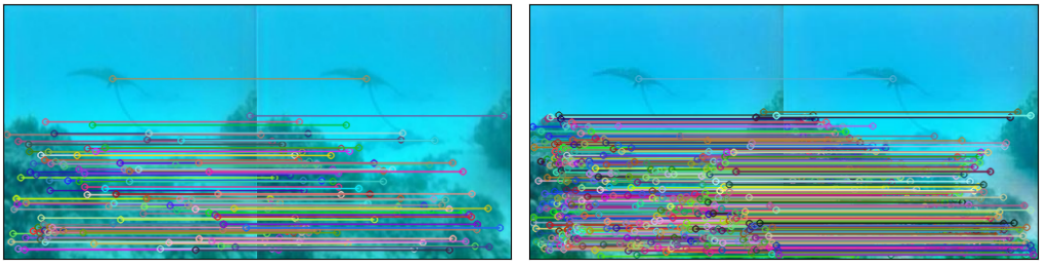
The intensity of the red channels for every pixel in the original noisy image tends to be mostly very low, whereas the blue and green channels are of great quantity. Enhancement balances the distribution, incrementing the intensity of red values and diminishing the one of the bluish channels. This is because quality recovery restores the RGB values and aims to remove the predominancy of lower-wavelength colors that create the redundant layer of backscatter.

5.3.3 SIFT Keypoints Matching

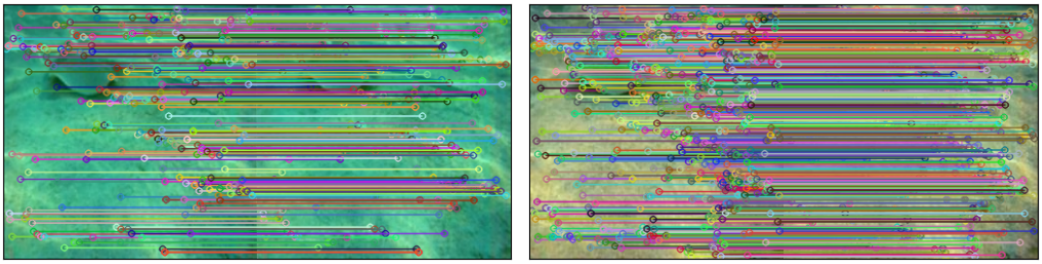
SIFT stands for Scale-Invariant Feature Transform, and it's used to find scale-invariant interest points (also called descriptors) in images. It's essential for many computer vision tasks such as object recognition, path detection, 3D reconstruction and image search. For the sake of the thesis, SIFT matching was performed on degraded and enhanced images to compare how many interesting points are located and found to study how impactful the model can be on the description of important keypoints of objects within the images.



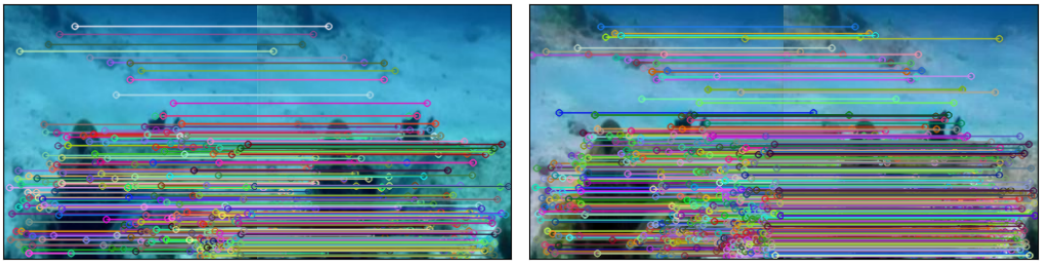
(a)



(b)

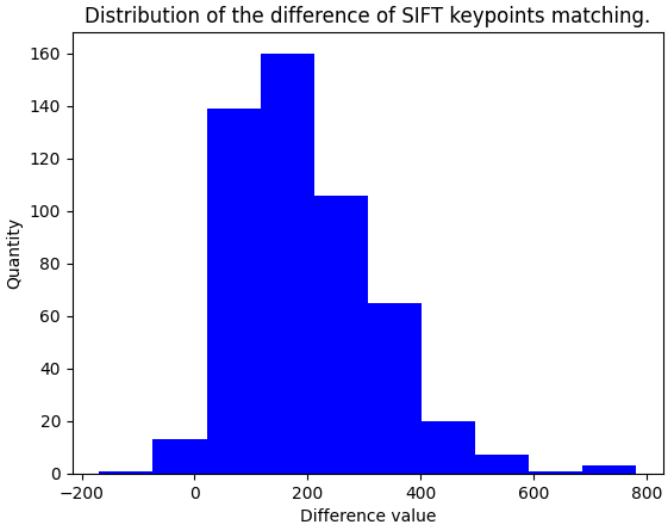


(c)



(d)

As can be noticed in the example images and in Table **5.3.3**, the model is able to detect a huge quantity of SIFT keypoints, leading to the discovery of more descriptors. This means that, in object detection tasks or in image retrieval, enhanced images would be more informative and could lead to more meaningful results. The total average of the two classes of images is computed on the whole EUVP test dataset, composed of 515 images. The results show the presence of a considerable difference between the descriptors found in degraded images and in images the model enhanced. The histogram of the distribution of the differences between the SIFT keypoints detected in all the enhanced images and all the degraded ones in the test set is plotted below. It shows that the enhancement and restoration algorithm mostly leads to an increment of 180-300 detected keypoints in enhanced images more than in degraded ones. This implies a substantial increase in the probability of locating objects and identifying structures in underwater environments.



	a	b	c	d	e	f	g	h	i	l	Average	**Total Average**
Degraded	224	129	291	250	662	872	431	228	992	163	424.2	**642.91**
Enhanced	568	454	623	431	934	1167	765	392	1477	333	714.4	**838.06**

Table 5.4. Results of SIFT matching on 10 random test images, their average and total average on the whole test set (515 images).

5.4 Effect on High-Level Vision Tasks

After qualitatively demonstrating how enhancement facilitates the identification of edges, colors, and keypoints, an additional experiment was conducted to evaluate the algorithm's performance in high-level vision tasks. Specifically, underwater divers 2D pose estimation was chosen as the focal task for this evaluation, that is one of the most challenging high-level vision tasks. A PyTorch implementation of the OpenPose[3] model was used: it jointly detects human body, hand, facial, and foot keypoints (in total 135 keypoints) on single images and was tested on multiple images of the Underwater Image Enhancement Benchmark (UIEB) Dataset depicting divers in varied underwater environments.

Four pictures were selected from *UIEB* dataset and resized to a resolution of 256x256 pixels. The results of this evaluation are depicted below: the left column showcases the original degraded underwater images prior to enhancement, while the right column exhibits the images after the application of the enhancement model.

[3]OpenPose, available at: `https://github.com/CMU-Perceptual-Computing-Lab/openpose`

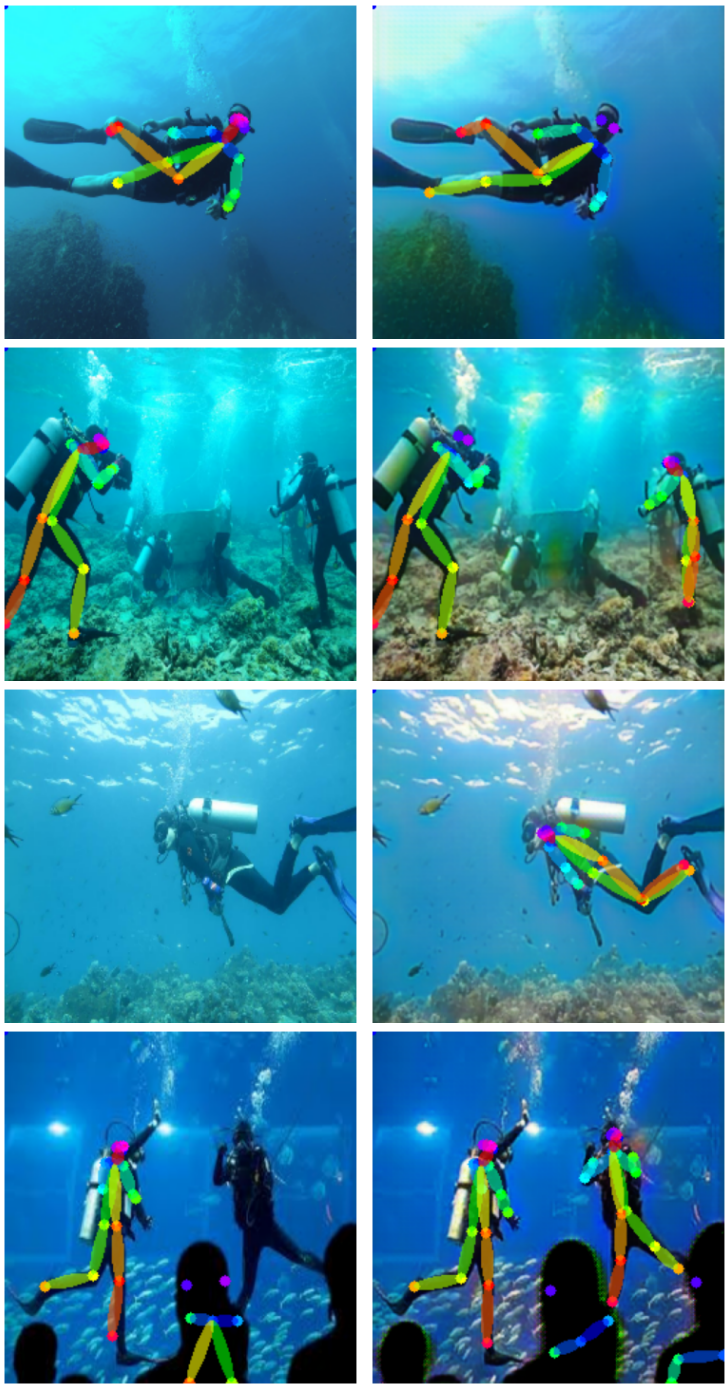


Figure 5.6. OpenPose pose detection algorithm on degraded underwater images (left) and enhanced underwater images (right).

The application of the OpenPose pose detection algorithm to divers images shows a notable improvement in the detection of human body postures and accurate human body joints with respect to its application on noisy underwater pictures.

DeformUGAN, being able to restore edges and contrasts, along with colors, allows the model to accurately detect figures and limbs. This leads to think that the algorithm would be very effective in other high-level vision tasks such as object detection or segmentation.

5.5 Limitations and Failure Cases

DeformUGAN is able to overall enhance and restore images taken in various underwater environments at different levels of degradation and with different situations. Nevertheless, some limitations have to be exposed in order to better understand how the network works and to lay a foundation for future studies. First of all, DeformUGAN is considerably computation demanding, since it required nearly 24 hours to be trained on the whole EUVP dataset with a batch size of 6. In addition, the model struggles to restore severely degraded and texture-less images, leading to inconsistent results. Sometimes the generated images appear oversaturated and the model tends to add redness in areas in which it isn't required.

Figure 5.7. Examples of failure cases: oversaturation (first two pictures) and severe degradation (last two pictures).

It's necessary to say that problems regarding the recovery of seriously degraded underwater images arised only when trying to enhance pictures from the UIEB Dataset, which the model wasn't trained on. From this, it's deemed appropriate to think that, with an accurate fine-tuning of DeformUGAN on such dataset, a substantial improvement in results could be achieved.

Conclusions

This thesis has addressed the problem of underwater image quality recovery through both image enhancement and image restoration, leveraging the power of these two methods in a unique framework. Firstly, a deep study of ocean environments and underwater optical imaging has been conducted, highlighting the causes of imaging degradation and describing the major attributes to be considered when performing image restoration. The second chapter has been dedicated to a comprehensive review of the state-of-the-art algorithms for the task, together with the existing datasets and the specific metrics regarding underwater image quality evaluation. After having described the architecture and mathematics behind the main AI structures used in the work, the proposed model was presented. Such model takes the name of DeformUGAN, and it's a conditional encoder-decoder GAN-based model that performs underwater image enhancement and restoration via modulated deformable convolution and wavelength-based residual connections and a SLIC-based loss function which aims to optimize the difference between the estimated atmospheric light of the generated image and the ground truth one. The proposed model also formulates the loss function by evaluating global similarity and image content. Deformable convolution is used in the downsampling part of the encoder and the discriminator of the network is a 4-layer Markovian PatchGAN architecture. The model was trained on the EUVP dataset, a large-scale dataset containing a collection of 11k paired underwater images for supervised training. Evaluation has been conducted in several steps: first of all, a broad ablation study demonstrated the impact of the introduced novelties (deformable convolutional layers, adaptive residuals, and atmospheric light loss function) on the overall performance of the model, showcasing how the model is able to learn muticontextual information and effective sampling locations. Then, the training setup was described, along with the experiments that lead to the final hyperparameters configuration. In particular, the model was trained on different batch sizes, SLIC cluster centers, and epochs and the best results were compared to other state-of-the-art works on image enhancement and restoration.

This quantitative comparison demonstrated that DeformUGAN outperforms state-of-the-art (SOTA) models in two specific metrics (PCQI and UCIQE) and consistently achieves positive results also over the other ones, often securing the second or, at most, the third-best value among the other works. A qualitative evaluation was then conducted to demonstrate that enhanced images effectively enrich the edge and feature information, useful for applications in advanced vision tasks. In fact, the model results were eventually tested on a 2D pose estimation task and demonstrated that enhancement and restoration really help on detecting human bodies with greater accuracy.

DeformUGAN has some limitations due to its computational complexity, and it sometimes fails with really degraded environments. For this reason, future developments have to be investigated. Unpaired training can definitely be considered for improving efficiency and inference time. The usage of smaller but fair datasets can also be a solution, together with the possibility of training with images of a bigger size than the one considered in the work (256x256). Moreover, adaptive learning can be further enhanced by using deformable convolution or exploring the wavelength-dependency in deeper architectures. Finally, the adoption of atmospheric light estimation can advance through transmission and IFM computation.

This thesis has set itself as a starting point for underwater image enhancement methods that seek to achieve the objective both through perceptual RGB pixel data recovery and through physics-based degradation restoration. Being the ocean a vast, powerful, but still nearly undiscovered world, future improvements will allow explorations to delve deeper into its mysteries, allowing marine research to progress for the preservation of our planet's most precious ecosystem.

Bibliography

[1] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34** (2012), 2274. doi:10.1109/TPAMI.2012.120.

[2] Akkaynak, D. and Treibitz, T. Sea-thru: A method for removing water from underwater images. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1682–1691 (2019). doi:10.1109/CVPR.2019.00178.

[3] Akkaynak, D., Treibitz, T., Shlesinger, T., Loya, Y., Tamir, R., and Iluz, D. What is the space of attenuation coefficients in underwater computer vision? In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 568–577 (2017). doi:10.1109/CVPR.2017.68.

[4] Anwar, S., Chongyi, L., and Porikli, F. Deep underwater image enhancement. (2018).

[5] Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan (2017). arXiv:1701.07875.

[6] Basodi, S., Ji, C., Zhang, H., and Pan, Y. Gradient amplification: An efficient way to train deep neural networks. *Big Data Mining and Analytics*, **3** (2020), 196. doi:10.26599/BDMA.2020.9020004.

[7] Chao-Fong, L. and Henley, J. Titanic sub live updates: source of 'banging noises' still unknown, says us coast guard, as search continues. *The Guardian.*, (2023).

[8] Chen, X., Zhang, P., Quan, L., Yi, C., and Lu, C. Underwater image enhancement based on deep learning and image formation model (2021). arXiv:2101.00991.

[9] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. Deformable convolutional networks. *CoRR*, **abs/1703.06211** (2017). Available from: http://arxiv.org/abs/1703.06211, arXiv:1703.06211.

[10] Doornekamp, R. The difference between rovs and auvs (2021). Available from: https://www.deeptrekker.com/news/difference-between-rovs-and-auvs.

[11] Fabbri, C., Islam, M. J., and Sattar, J. Enhancing underwater imagery using generative adversarial networks. In *IEEE International Conference on Robotics and Automation (ICRA)* (2018).

[12] Fu, X., Liang, Z., Ding, X., Yua, X., and Wang, Y. Image descattering and absorption compensation in underwater polarimetric imaging. *Optics and Lasers in Engineering*, **132** (2020). Available from: https://www.sciencedirect.com/science/article/pii/S0143816620300403, doi:https://doi.org/10.1016/j.optlaseng.2020.106115.

[13] Guan, Y. e. a. Fast underwater image enhancement based on a generative adversarial framework. *Frontiers in Marine Science*, (2023).

[14] Hassan, N. e. a. The retinex based improved underwater image enhancement. *Multimedia Tools and Applications*, (2020).

[15] He, K., Sun, J., and Tang, X. Single image haze removal using dark channel prior. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1956–1963 (2009). doi:10.1109/CVPR.2009.5206515.

[16] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *CoRR*, **abs/1512.03385** (2015). Available from: http://arxiv.org/abs/1512.03385, arXiv:1512.03385.

[17] Hecht-Nielsen, R. Theory of the backpropagation neural network. In *International 1989 Joint Conference on Neural Networks*, pp. 593–605 vol.1 (1989). doi:10.1109/IJCNN.1989.118638.

[18] Islam, M. J., Xia, Y., and Sattar, J. Fast underwater image enhancement for improved visual perception. *IEEE Robotics and Automation Letters*, **5** (2020).

[19] Islam, M. J. e. a. Semantic segmentation of underwater imagery: Dataset and benchmark. *Something Journal*, (2020).

[20] Islam, M. J. e. a. Simultaneous enhancement and super-resolution of underwater imagery for improved visual perception. (2020).

[21] Isola, P., Zhu, J., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. *CoRR*, **abs/1611.07004** (2016). Available from: `http://arxiv.org/abs/1611.07004`, `arXiv:1611.07004`.

[22] Jerlov, N. G. *Marine Optics*. Elsevier, Amsterdam (1976).

[23] Jingchun, Z., Tongyu, Y., and Weishi, Z. Underwater vision enhancement technologies: a comprehensive review, challenges, and recent trends. *Applied Intelligence*, (2022). (2023).

[24] Li, C. e. a. An underwater image enhancement benchmark dataset and beyond. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, (2019).

[25] Li, H., Li, J., and Wang, W. A fusion adversarial underwater image enhancement network with a public test dataset. (2019).

[26] Li, J. e. a. Watergan: Unsupervised generative network to enable real-time color correction of monocular underwater images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2020).

[27] Li, M., Liu, J., Yang, W., Sun, X., and Guo, Z. Structure-revealing low-light image enhancement via robust retinex model. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, (2018).

[28] Marques, T. P., Albu, A. B., and Hoeberechts, M. A contrast-guided approach for the enhancement of low-lighting underwater images. *Journal of Imaging*, (2019).

[29] Midjourney. Midjourney (2022). [Online]. Available from: `https://www.midjourney.com/home?callbackUrl=%2Fexplore`.

[30] Mirza, M. and Osindero, S. Conditional generative adversarial nets. *CoRR*, **abs/1411.1784** (2014). Available from: `http://arxiv.org/abs/1411.1784`, `arXiv:1411.1784`.

[31] O'Shea, K. and Nash, R. An introduction to convolutional neural networks. *CoRR*, **abs/1511.08458** (2015). Available from: `http://arxiv.org/abs/1511.08458`, `arXiv:1511.08458`.

[32] Panetta, K., Gao, C., and Agaian, S. Human-visual-system-inspired underwater image quality measures. *IEEE JOURNAL OF OCEANIC ENGINEERING*, (2016).

[33] Panetta, K., Zhou, Y., and Wharton, E. Parameterized logarithmic framework for image enhancement. *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS*, (2011).

[34] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents (2022). `arXiv:2204.06125`.

[35] Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, **abs/1505.04597** (2015). Available from: `http://arxiv.org/abs/1505.04597`, `arXiv:1505.04597`.

[36] Schechner, Y. and Karpel, N. Clear underwater vision. (2004).

[37] Schechner, Y. and Karpel, N. Recovery of underwater visibility and structure by polarization analysis. *IEEE JOURNAL OF OCEANIC ENGINEERING*, (2005).

[38] Schettini, R. and Corchs, S. Underwater image processing: State of the art of restoration and image enhancement methods. (2010).

[39] Sharma, P. K., Bisht, I., and Sur, A. Wavelength-based attributed deep neural network for underwater image restoration (2022). `arXiv:2106.07910`.

[40] Smitha, R., Mukesh, D. P., and Gajanan, K. B. Underwater image enhancement: a comprehensive review, recent trends, challenges and applications. *Artificial Intelligence Review*, (2021).

[41] Society, N. G. Ocean (2023). [Online; in data 19-ottobre-2023]. Available from: `https://education.nationalgeographic.org/resource/ocean/`.

[42] Srinivas, S. e. a. Channel prior based retinex model for underwater image enhancement. (2022).

[43] Stutz, D., Hermans, A., and Leibe, B. Superpixels: An evaluation of the state-of-the-art. *CoRR*, **abs/1612.01601** (2016). Available from: `http://arxiv.org/abs/1612.01601`, `arXiv:1612.01601`.

[44] Sun, B., Mei, Y., Yan, N., and Chen, Y. Umgan: Underwater image enhancement network for unpaired image-to-image translation. *Journal of Marine Science and Engineering*, (2023).

[45] Tan, C., Seet, G., Sluzek, A., and He, D. A novel application of range-gated underwater laser imaging system (ulis) in near-target turbid medium. *Optics and Lasers in Engineering*, (2004).

[46] Tang, P. e. a. Real-world underwater image enhancement based on attention u-net. *Journal of Marine Science and Engineering*, (2023).

[47] Tang, S., Li, C., and Pan, X. A simple illumination map estimation based on retinex model for low-light image enhancement. *Journal of Something*, (Year).

[48] Tian, T., Cheng, J., Wu, D., and et al. Lightweight underwater object detection based on image enhancement and multi-attention. *Multimedia Tools and Applications*, (2024). doi:10.1007/s11042-023-18008-8.

[49] Treibitz, T., Neal, B., and Kline, D. e. a. Wide field-of-view fluorescence imaging of coral reefs. *Sci Rep*, **5** (2015), 7694.

[50] Wang, I. Y. e. a. Ba-gan: Block attention gan model for underwater image enhancement. In *IEEE International Conference on Robotics and Automation (ICRA)* (2021).

[51] Wang, Y., Guo, J., Gao, H., and Yue, H. Uiec^2-net: Cnn-based underwater image enhancement using two color space. (2021).

[52] Wang, Y., Zhang, J., Cao, Y., and Wang, Z. A deep cnn method for underwater image enhancement (2017).

[53] Wu, H., Qu, Y., Lin, S., Zhou, J., Qiao, R., Zhang, Z., Xie, Y., and Ma, L. Contrastive learning for compact single image dehazing. *CoRR*, **abs/2104.09367** (2021). Available from: https://arxiv.org/abs/2104.09367, arXiv:2104.09367.

[54] Yang, M. and Sowmya, A. An underwater color image quality evaluation metric. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, (2015).

[55] Zhang, L., Wang, S., and Wang, X. Single image dehazing based on bright channel prior model and saliency analysis strategy. *IET Image Processing*, **15** (2021), 1023. doi:10.1049/ipr2.12082.

[56] Zhang, M. and Peng, J. Underwater image restoration based on a new underwater image formation model. *IEEE Access*, **6** (2018), 58634. doi:10.1109/ACCESS.2018.2875344.

[57] Zhang, Q., Xiao, J., Tian, C., Lin, J. C.-W., and Zhang, S. A robust deformed convolutional neural network (cnn) for image denoising. *CAAI Trans. Intell. Technol.*, **8** (2022), 331. Available from: https://api.semanticscholar.org/CorpusID:249797209.

[58] Zhang, W. e. a. Color correction and adaptive contrast enhancement for underwater image enhancement. *Computers & Electrical Engineering*, **91** (2021).

[59] Zheng, M. and Luo, W. Underwater image enhancement using improved cnn based defogging. (2022).

[60] Zhou, J., Zhang, D., and Ren, W. Auto color correction of underwater images utilizing depth information. *IEEE GEOSCIENCE AND REMOTE SENSING LETTERS*, (2022).

[61] Zhu, X., Hu, H., Lin, S., and Dai, J. Deformable convnets v2: More deformable, better results. *CoRR*, **abs/1811.11168** (2018). Available from: http://arxiv.org/abs/1811.11168, arXiv:1811.11168.