Journal of Computer Science

Review

Denoising of Underwater Images with Regulated Autoencoders

Shreya Priyadarshini Roy, Sivagami M. Angelin Beulah S. and Maheswari N.

School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India

Article history
Received: 21-11-2023
Revised: 09-03-2024
Accepted: 13-03-2024

Corresponding Author:
Sivagami M.
School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India
Email: msivagami@vit.ac.in

Abstract: In the domain of underwater image processing, the images obtained in underwater lighting conditions are of very poor in quality and contain distortions due to the scattering of lights and color absorption of images in water. This makes it difficult to analyze these images for scientific, commercial and military purposes. Therefore, there is a need for efficient techniques that can remove the noise from these images. In this study, the experimental study of regulated autoencoders namely, denoising autoencoders, sparse autoencoders, deep autoencoders, variational autoencoders and convolutional autoencoders for noise removal in underwater images have been performed and the. These models have been implemented in Python, trained tested and evaluated on the dataset known as Enhancing Underwater Visual Perception (EUVP) image dataset. Our results show that Denoising Autoencoder (DAE) can effectively remove noise from underwater images, with a good structural index and high peak signal-to-noise ratio. The sparse autoencoder was fine-tuned to fit for underwater image denoising. However, our study shows that the obtained results show that Denoising Autoencoder (DAE) can enhance the quality of underwater images visually and effectively denoise the images.

Keywords: Autoencoder, Convolutional Autoencoder, Deep Autoencoder, Denoising Autoencoder, PSNR-Peak Signal to Noise Ratio, Sparse Autoencoder, SSIM-Structural Similarity Index Measure, Variational Autoencoder

Introduction

Underwater image processing plays a vital role in the real world, in studying the pattern and behavior of the marine world. Due to light absorption and scattering in aquatic environments, underwater images generally suffer from low contrast, poor visibility and color distortion. With technological improvements, underwater image study has become a good field of study for researchers who would ensure to preservation of the underwater ecosystem. The marine ecosystem gives us an insight into the marine habitat and vegetation but also help a lot in scientific research as well. Underwater environments are subjected to poor lighting conditions, water turbidity, camera sensors that were used to capture the pictures, underwater motions and blurring effects. So eventually, there could be various forms of noises and distortions in the image that is captured. Efficient underwater image denoising is critical for various applications like marine biology, underwater exploration research and archaeology. Image noises refer to undesired fluctuations in color that obscure details in the captured image. Noise in the image may arise from underexposure, when pixels in the intended image have little light variation to report but are being over-amplified by higher ISO settings. In addition, a variety of other problems, such as heat from the sensors or outside interference, can make sensors vulnerable and cause noise in the final image. The ways that light scatters and absorbs in water influence the overall performance of underwater imaging systems. Additional substances like dissolved organic materials or tiny visible floating particles can also have absorption and scattering effects in addition to water. The effects of absorption and scattering are amplified by the presence of floating particles, often known as marine snow (Hou *et al*., 2015).

Artificial lighting can expand the visibility range, but it tends to illuminate the environment unevenly, creating a brilliant spot in the middle of the image with a badly lighted area surrounding it. Additionally, depending on their wavelengths, colors disappear one by one as the amount of light diminishes with depth. The limited range visibility and contrast with uneven lighting and blurring, the bright artifacts, decreased colors and noise effects can all affect the underwater images. For using the underwater images, it is crucial to eliminate these disturbances. Underwater images can be further used for studying underground volcanoes to tectonic plate activities. It is believed that most of the prehistoric earth data is preserved under the sea. Images are the only way most researchers have access to that data. Hence by enhancing the images not only the researchers can be benefitted but also, but also many mysteries of the earth we live in can be understood.

Image restoration aims to improve the visual quality of an image by reconstructing the original image which was a degraded or noisy image. This can be achieved with image enhancement filters, designed to minimize the effects of various degradations such as noises due to sensors, blur effects, inappropriate focus of the camera, relative motion of the camera and atmospheric pressure. Unlike image enhancement, which is subjective relying on the user's preferences, image restoration is an objective process based on mathematical or probabilistic models of image degradation. In general practice, the images that were denoised can be denoised using median filters, gaussian blur filters, etc. The dataset that has been pre-processed using various filters can be fed into an autoencoder for denoising.

An autoencoder disregards the signal noise and creates an input representation for the given input data by reducing the dimensionality of the input. It is a type of unsupervised neural network to learns and identifies the crucial patterns in an image. A reconstruction side is also taught along with the reduction side and it involves an autoencoder to attempt and create an image representation by minimizing the encoding data and creating an output like the input image without the noise. This enables autoencoders to detect crucial data points. A denoising autoencoder is a type of autoencoder that does not allow the neural network to learn the identity function of the input image. The implementation of autoencoders has many issues, including a lack of training data, training for the incorrect use case, too lossy and imprecise decoding, misinterpretation of critical variables and an inadequate bottleneck layer. Hence, all these must be kept in mind before using autoencoders, the purpose of a denoising autoencoder is not to resemble the identity function. Defective data can be recovered using a denoising autoencoder. It may be employed as a feature selection tool because it reduces noise by filtering it out.

Contribution of the Paper

The main contributions of the work are:

- Detailed experimental study on five different types of regulated autoencoders such as denoising autoencoder, sparse autoencoder, deep autoencoder, variational autoencoder and convolutional autoencoders has been done on underwater images and have concluded that the Denoising Autoencoders (DAE) is best suited for noise removal in underwater images based on the PSNR and SSIM metrics
- The sparse autoencoder was not appropriate for underwater images. So, it has been fine-tuned in such a way as to work for underwater images by adding dropout layers. Though the sparse autoencoder has been modified and fine-tuned, the Denoising Autoencoder (DAE) still had a high PSNR value and works best in noise removal. The details have been briefed in the sections below

The paper has been written in such a way the literature review section discusses the existing techniques available for the enhancement of underwater images. The dataset information and the experimental setup are discussed in the next section. The experimental study of autoencoders with under-water images in which the structure of each of the five different types of autoencoders for denoising the image dataset namely denoising autoencoder, sparse autoencoder, deep autoencoder, convolutional and variational autoencoder have been discussed elaborately in the further sections. The sparse autoencoder was also tuned so that denoising could be done on underwater images by making modifications to the layers. The details are briefed in the sparse autoencoder section along with the validation details that have been considered. The results that have been obtained for PSNR and SSIM for the five autoencoders and the output images for the five autoencoders are discussed in the result analysis section. The conclusions are based on our outputs obtained and the future directions that could be done from this study are elaborated.

Over the past years, underwater images have been analyzed and many researchers have been trying to enhance the image resolution and contrast and remove the noise from underwater images. A survey on denoising techniques over the past years has been done and briefed in this review.

Dabov *et al*. (2007) demonstrated an image-denoising method based on an improvised sparse representation in the transformation domain. The results of the experiments showed that this approach performs better than most algorithms. Denoising 1-D signals, video and pictures were the few examples of applications where the recommended method might be modified to make use of exceedingly sparse signal representations.

Kalantari *et al*. (2015) proposed a nonlinear regression model, based on a multilayer perceptron neural network, to understand the intricate link between optimum filter parameters and noisy scene data. The trained network produced filtered images quickly with different effects like depth of field, blurred motion, area illumination and glassy reflections. The authors had done noise removal in digital images using a non-linear PDE-based algorithm.

The results showed that the noise present has been largely removed while keeping the edges intact. Overall, the authors have highlighted the potential of machine learning techniques for generating high-quality filtered images in a short time frame.

The researcher in his work Gondara (2016) had implemented Convolutional Denoising Autoencoders (CDAE) for medical image denoising. The approach was based on a deep learning architecture that learned from large datasets and extracted the features that were essential for denoising. Traditional denoising techniques were compared to the CDAE approach and the results showed that the CDAE methodology outperforms existing techniques in terms of both objective and subjective evaluations. The proposed approach for medical image denoising was useful for enhancing image quality and facilitating diagnosis.

Skribtsov and Surikov (2016) have suggested a regularization method for training sparse denoising autoencoders for designing a model for image denoising and inpainting. This model gave greater stability to the denoising methods. The method processed all the images gathered from various types of sensors such as visible spectrum and multispectral sensors. Small-sized objects and identical areas can be effectively stripped of noise using linear and nonlinear filters, but not applicable to heterogeneous areas. To address this limitation (Jeelani and Veena, 2017) used an optimum filter that utilized locally estimated parameter values for noise removal. The proposed filter outperformed other well-known filters in terms of noise reduction, as evaluated by MSE and PSNR metrics. Overall, the research highlighted the importance of considering the local characteristics of an image for developing effective image-denoising filters.

Iqbal *et al*. (2007) the objective was to enhance image quality. First, the RGB contrast stretching algorithm is used. Second, to address the lighting issue, saturation and intensity stretching are applied. The development of interactive software for underwater image enhancement. The benefit of using two stretching models is that it can assist balance out lighting issues and color issues. In Xiang and Pang's (2018) work an improved Denoising Autoencoder (DAE) for image denoising was proposed. The proposed model incorporates both local and non-local information through an encoder-decoder architecture. The encoder extracts local features while the decoder learns non-local features using a self-attention mechanism. The authors evaluated their proposed method against several state-of-the-art denoising techniques and demonstrated its superior performance. They also carried out ablation research to demonstrate the efficiency of every part of the suggested model. Overall, the study offered an efficient deep learning-based picture-denoising approach.

Fan *et al*. (2019) presented a summary of significant research in image denoising, highlighting various techniques and their characteristics. As the complexity and demands of image denoising continue to grow, noise analysis has become an increasingly important tool for developing new denoising methods. In the study by Hashisho *et al*. (2019), it was suggested to use a U-net denoising autoencoder for underwater image color restoration. It discussed the challenges in underwater imaging and how the proposed method addresses them. Pre-processing, feature extraction, denoising and color restoration were all done and the results showed that the U-net denoising autoencoder was a cutting-edge technique in image denoising.

Han *et al*. (2018) and the authors discussed the cause of underwater image degradation and examined the strategies used for underwater image dehazing and restoration using deep learning techniques. The applications of underwater image processing were briefed and underwater image color evaluation metrics were elaborated. A complete survey of all the methods proposed had been done, including their shortcomings. Goyal *et al*. (2020) did a complete comparison and evaluation of various image-denoising methods. Spatial domain, transform domain, hybrid, sparse representation and dictionary learning methods are the five groups in which underwater image denoising techniques have been divided. These techniques have higher noise levels. To remove the noise the study suggested that combining different denoising methods from various domains can help overcome the limitations of individual methods and harness their unique attributes. A single denoising algorithm may not be sufficient to achieve optimal results.

Zhou *et al*. (2020) in their work have suggested the use of the Stacked Convolutional Sparse Denoising Autoencoder (SCSDA) model to eliminate noise from data containing heterogeneous undersea information. To uncover the underlying features of the data and eliminate noise, the model made use of the characteristics of Convolutional Neural Networks (CNNs) and sparse coding. SCSDA model is compared with several existing denoisers on simulated and real-world underwater images and the results demonstrated its superior denoising performance. The proposed model has potential applications in underwater image processing and other fields where heterogeneous information data needs to be denoised.

The noise data in the GPR images were not uniform, so a fixed threshold in the denoising algorithms in each sub-band of the hyper-wavelet cannot be used. To address this issue, a new denoising algorithm called NSST-based GWO has been developed. This algorithm used the Grey Wolf Optimization (GWO) method to optimize the threshold values and provided exceptional noise removal while preserving the image edges. The experimental results showed that the suggested approach performs better than the current denoising methods for GPR images. Therefore, effective GPR image denoising can be achieved using the suggested NSST-based GWO algorithm (Zhou *et al*., 2020).

For many types of noise, like Gaussian, impulse, Poisson, mixed and real-world noise, the application of image denoisers based on machine learning and Generative Adversarial Networks (GANs) is investigated. Based on their PSNR values, the denoisers are assessed and the best outcomes for various noise kinds are addressed along with prospects. A complete analysis of many image-denoising models based on machine learning for image denoising has been done. In the paper, Thakur *et al*. (2021) have provided valuable insights into the current state of machine learning-based image de-noising algorithms. Cherian *et al*. (2021) proposed a new model for de-noising that increased the quality of the image. CLAHE is applied to improve the contrast of the image furthermore. The reconstructed images had a high SNR value (Thakur *et al*., 2021).

Du Toit *et al*. (2023) proposed a deep supervised autoencoder model with an optimized Bayesian hyperparameter for terrain classification. The autoencoder was combined with a supervised learner to perform the classification of different types of terrain. After training and fitting the dataset over six distinct terrain surfaces-asphalt, dirt, epoxy, grass, paving and stone surfaces-with a raspberry Pi computer and a sense HAT inertial measurement unit, Bayes hyperparameter values were determined. The results were compared with an SVM classifier, a regression model and an XGBoost model and was found that the supervised autoencoder could be used effectively for terrain classification. Chandra *et al*. (2023) suggested and implemented a deep CNN-based Color Balancing and Denoising Technique (CNN-CBDT) with the activation function RELU to enhance the underwater images. The model has enhanced the images with good PSNR values and SSIM values. The PSNR value was increased to 17% of the original value of 19.580 and the SSIM was 15% enhanced with the value of 0.952. The proposed method.

Arbahri *et al*. (2024) have worked on oceanographic data for the prediction of critical marine resources using machine learning algorithms. The agency for the study and application of technology's (BPPT) marine database provided the oceanographic data for the years 2009-19. The raw data was converted with respect to the oceanographic parameters such as conductivity, temperature and depth of data. Supervised machine learning algorithms such as Decision Tree (DT), Linear Regression (LR) and random forest classifiers have been applied to the dataset and the outputs were compared and analyzed. This study has the potential to provide data and information that may be used as a reference for creative investigations.

Li *et al*. (2024) have proposed an adaptive underwater image enhancement method that uses a legitimate dataset with ground truth to minimize haze and correct image color. The hue channel statistics have been used to create a dataset of color-corrected images from different underwater images. A transformer-like network has been trained with a hazy terrestrial image dataset to remove the haziness. The model performed in a robust manner for underwater images when compared to other state-of-the-art methods.

From the literature, it is understood that various efforts have been taken to enhance the quality of the underwater images. The image processing methods are also used for removing the haziness and color distortions. In this era, numerous deep-learning models are now being exploited for the enhancement of marine images. Image processing methods may have limitations in providing the number of images to the model. Whereas the deep learning models can have an input of a huge dataset of images.

Dataset

The Enhancing Underwater Visual Perception (EUVP) an online largescale dataset has been used in this study. This underwater image dataset is designed to make the training of underwater image enhancement models easier. It includes paired and unpaired image samples with different perceptual qualities. The paired dataset is prepared with 11670 underwater dark images, 8670 ImageNet and 4500 underwater scenes. The unpaired dataset consists of 6665 images of which good-quality images are 3140 and poor-quality images are 3195. The images were downloaded in the google drive and were kept ready for applying pre-processing techniques. The dataset has a collection of 20 k images which was used in our work. The dataset is available at http://irvlab.cs.umn. edu/resources/euvp-dataset.

Training Model

To train an image dataset with an autoencoder various issues were faced. Images are typically high-dimensional data, which means that each image can have many pixels, resulting in a high number of input features. This makes it computationally difficult and the time taken to train the autoencoder is very high leading to overfitting with image datasets. This may require the use of distributed computing resources, such as cloud-based services or a cluster of GPUs. When a model is overfitting, it gets exceedingly complex and begins to memorize the training set of data rather than learning broad traits that may be applied to fresh data.

Pre-processing the image data is a complex task, as it may involve tasks such as resizing, normalization and augmentation. Choosing the right pre-processing

DOI: 10.3844/jcssp.2024.670.681

techniques can influence the performance of the autoencoder. Converting an image dataset into a NumPy dataset is an essential step in building an autoencoder model for image reconstruction. However, this process can be challenging, requiring attention to detail to ensure the model can accurately learn the features of the dataset. The versions of NumPy, OpenCV and Matplotlib libraries were installed and must be compatible. Loading the image dataset with OpenCV is another challenge, requiring images to be in the correct format. The images must also be pre-processed to ensure effective learning, requiring techniques such as resizing, normalization and color space conversion. The optimal setting for each image requires experimentation.

Converting images into NumPy arrays requires the dimensions of the resulting array to correspond to the image size and number of channels. Saving the dataset must also ensure it is in a format easily loaded for processing and training the autoencoder. Using the NumPy library's 'Savez' function to save the dataset as a compressed archive file accomplishes this. By addressing each issue, the image dataset can be successfully converted into a NumPy dataset, preparing it for further processing and training of the autoencoder model.

Experimental Setup of Autoencoders with Under-water Images

Traditional image processing methods are often insufficient in removing noise and artifacts from underwater images, resulting in low visual quality. To address this, learning-based methods such as autoencoders have been deep extensively used for image denoising.

The experiments were done in google Co-lab. The dataset was loaded into google drive and to obtain reliable findings, the dataset had to be divided into both clear and noisy image samples. But the labeling was not sequential, hence it was converted into a numpy format. After manually labeling the pairs with their corresponding noisy and clear images the dataset had been converted into numpy format. While converting the images into numpy arrays loss of data occurs such as color depth, resolution, or pixel values and hence that needs to be taken care of. Before converting images into numpy arrays, they require preprocessing such as resizing, normalization and cropping to ensure that they are of uniform size and quality. This can add additional complexity and time to the conversion process.

Autoencoders

In the realm of unsupervised machine learning, autoencoders are a unique kind of neural network. From their compressed version, these autoencoders learn to recreate images, text and data. The input image is compressed or down-sampled by the encoder into a latent space representation, which is a compressed or warped replica of the original image. The decoder will then use this as input to rebuild the image to its original image dimension. This could be a lossy reconstruction of the original image. The autoencoder functions are as given below:

$$P: X^n \rightarrow \mathbb{X}^p \ (Encoder) \tag{1}$$

$$Q: \mathbb{X}^p \rightarrow X^n \ (Decoder) \tag{2}$$

These functions must satisfy the minimization of an expected function value:

$$arg\, min_{P.Q}\, E[\Delta(z, Q\ o\ P(z)] \tag{3}$$

In this case, the reconstruction loss function Δ, measures the discrepancy between the decoder's output and input and E is the expectation throughout the z distribution, are both present. "P" is the encoder function mapped to the latent space represented as "z," and "Q" is the decoder function mapped to the latent space represented as "z." Figure 1 shows the autoencoder model in action (Kalantari *et al*., 2015). The commonly used reconstruction loss for the autoencoders is Mean Squared Error (MSE) loss:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \tag{4}$$

Here N is the number of elements in the input data and x_i and $\hat{x}_i$ are the i^{th} elements of the input and reconstruction respectively. The reconstruction loss function is designed to minimize the difference between the input and the output of the autoencoder and thereby learns the representation of the data that captures its important features.

Figure 1 shows the general architecture of a regulated autoencoder. "*X*" is the input image which is encoded by the encoder to generate a compressed representation and the decoder reconstructs the image "*X*" from the compressed representation by learning only the important pixels and on the way removes the noise in the image.

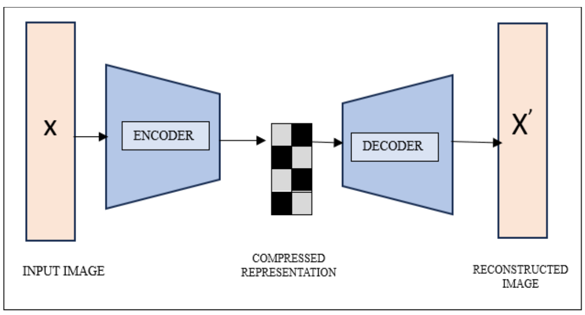

Fig. 1: Architecture diagram of autoencoder (source: Bank *et al*., 2023)

To overcome the drawbacks of undercomplete and overcomplete autoencoders, regularized autoencoders have been used. In this study, five different regulated autoencoder models such as denoising autoencoder, sparse autoencoder, deep autoencoder, convolutional autoencoder and variational autoencoder improve the visual quality of underwater images. The usage of these several autoencoder types enables an accurate evaluation of each model's efficiency in underwater image denoising. By selecting and comparing multiple models, this study aims to develop an approach that can provide high-quality denoising results for underwater images.

Denoising Autoencoder

Denoising the autoencoders model prevents simply copying the input to the output without picking up crucial details. This autoencoder adds some noise to a partially corrupted input to achieve this. During training, the model tries to rebuild the original, unaltered input from this damaged input. A mapping between the input data and a lower dimensional manifold that characterizes the natural data is learned to successfully cancel out the extra noise. The objective of this approach is to obtain a good representation of the input that can be reliably obtained from a corrupted input and used to recover the corresponding clean input. The lingering nodes that remain, duplicate the given input to a noisy input and randomly skew a portion of the input values. The model then attempts to minimize the loss function between the corrupted input and the output node (Hou *et al*., 2015).

Formally, let us say that $(\tilde{x}) \sim p$, $(\tilde{x}) \backsim p\ (\tilde{x}|x)$ jointly define a conditional probability of distribution $p\ (x, \tilde{x}) = p\ (\tilde{x}|x)\ p(x)$. A posterior distribution representation is given by a denoising autoencoder. The numerator represents the joint probability of observing $\tilde{x}$ given x multiplied by the prior probability of x. The denominator assures that the conditional probability distribution $p\ (\tilde{x}|x)$ is normalized over all values of x for the given $\tilde{x}$:

$$p(x|\tilde{x}) = \frac{p\ (\tilde{x}|x)p(x)}{\int_{X}^{\alpha} p(\tilde{x}\ |y)p(y)dy} \tag{5}$$

Figure 2 depicts the architecture of a denoising autoencoder. An underwater image is given as input and a random noise is added to the image to create a noisy image. The noisy image is given as input to the encoder which consists of convolution layers that generate an encoded representation.

The encoded representation is given to the decoder which has the transposed convolution layers. The decoder generates the noise-free image as output. The denoising autoencoder is the latest and most effective denoising technique that has been implemented for our EUVP dataset.

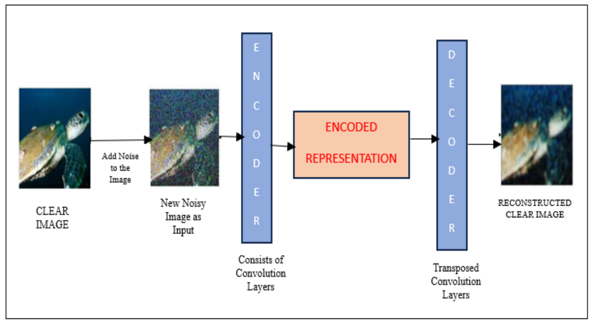

Fig. 2: Denoising autoencoder architecture

Sparse Autoencoder

Sparse autoencoders are neural networks that have more hidden nodes than input nodes, through which they extract key features from the input data. A sparsity restriction is added to the hidden layer to make sure the output layer does not just replicate the input data. This can be implemented by either manually setting all but the strongest activations to zero or by manually introducing additional terms to the loss function during training that promote a low probability distribution of hidden unit activations. Sparse autoencoders are designed to address the problem of overfitting by introducing a sparsity penalty on the hidden layer. In addition to the reconstruction error, a small but not quite zero amount is added as the sparsity penalty. To avoid employing all the hidden nodes at once, this promotes the model to use fewer hidden nodes overall.

The larger activation values in the hidden layer are alone considered by sparse autoencoders, while the remaining activation values are set to zero. The model's efficacy is increased because of this strategy, which limits the model's reliance on a small subset of the hidden nodes. The trained model's individual nodes must be data-dependent for this method to work, which means that various inputs will cause different nodes to be active throughout the network. Taking into consideration the scenario in which we have two loss functions, L_1 and L_2 stand for L_1 and L_2 regularisation, respectively.

$$L_1 = \parallel w \parallel, L_2 = w^2 \tag{6}$$

The L_1 regularization used in sparse autoencoder and the loss function are given as below:

$$Obj = L(x, \hat{x}) + regularization + \lambda \textstyle\sum_i |a_i^{(h)}| \tag{7}$$

In addition to the first two terms, add a third term that degrades the absolute value of the activation vector for sample "i" in layer h. The effect on the entire loss function is then controlled using a hyperparameter. Our construction of a sparse autoencoder follows.

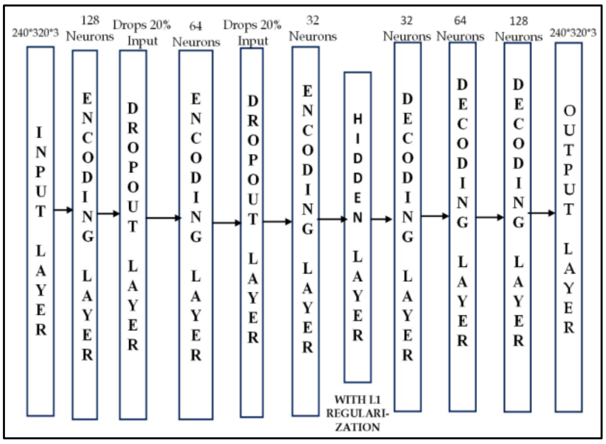

Fig. 3: Sparse autoencoder architecture

Figure 3 shows the architecture of a sparse autoencoder. Here the general architecture has been modified by adding dropout layers which drop 20% of the input. The hidden layer has a loss function of L_1 regularization. This is a variation from the general architecture where there will be only encoding, decoding and hidden layers. Adding the dropout layers removes the issue of overfitting during the training process.

Adding Drop-Out Layer in Sparse Autoencoder

During the creation of the sparse autoencoder model for fitting the EUVP dataset, some modifications were made to the original model. One important addition was the incorporation of dropout layers to help alleviate the issue of overfitting during training. Following each encoding layer, a dropout layer was introduced, which performs a dropout operation where 20% of the neuron outputs are randomly set to zero. The dropout layer's goal is to increase the model's capacity for generalization by preventing individual neurons from unduly depending on certain characteristics or patterns in the training data. By randomly dropping out a portion of the neuron outputs during each training iteration, dropout introduces a form of regularization that encourages the neurons to learn more robust and diverse representations of the data.

In the sparse autoencoder model, the dropout layers were strategically placed after each encoding layer. This positioning allowed for dropout to be applied to the encoded representations at multiple levels, ensuring that the learned sparse features were not overly influenced by any single layer. By applying dropout layers, the model became more resilient to overfitting, which is especially important when dealing with limited or noisy images such as the EUVP dataset. By introducing dropout layers with a 20% dropout rate, the model effectively reduced the risk of overfitting and improved its ability to generalize to unseen data. This adjustment helped enhance the performance and robustness of the sparse autoencoder model for denoising the underwater images in the EUVP dataset.

Deep Autoencoder

A Deep Autoencoder is constructed with two alike networks, the encoder and the decoder which both combine to form an unsupervised neural network. The deep encoder is pre-trained layer-by-layer with a Restricted Boltzmann Machine (RBM) which is also an unsupervised learning network that learns the probability distribution over a set of input data in each layer. Deep belief networks are the core building blocks of the RBMs that are utilized to comprehend how data is dispersed. The dropout layer's purpose is to increase the model's generalizability by preventing individual neurons from becoming overly dependent on specific traits or patterns in the training data. These models have a wide range of applications, including topic modeling, which involves statistically modeling abstract ideas dispersed throughout a group of papers. Additionally, deep autoencoders can compress images into 30-number vectors. In cases where the input data is real-valued, Gaussian rectified transformations can be used for the RBMs. Since there are more model parameters than there are input data, overfitting may be a problem. Training deep autoencoders can also be challenging since the learning rate needs to be adjusted during the decoder's backpropagation based on the binary or continuous nature of the data. Slowing down the learning rate can help prevent overfitting. The definition of the convolution operation in the 2D discrete space:

$$O(i,j) = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} F(u,v) I(i-u, j-v)) \tag{8}$$

In the image domain where the signals are finite, this formula becomes:

$$O(i,j) = \sum_{u=-2k-1}^{2k+1} \sum_{v=-2k-1}^{2k+1} F(u,v) I(i-u, j-v) \tag{9}$$

Figure 4 depicts the architecture of a deep autoencoder. A deep autoencoder consists of a series of convolutional and max pooling layers before the encoding layer which later compresses the image to a compressed format and then the decoder up samples the image with the and thereby removes the noise.

The other challenges of deep autoencoder are computational complexity, vanishing or exploding gradients problems, dimensionality reduction and complex representations leading to complicated interpretability.

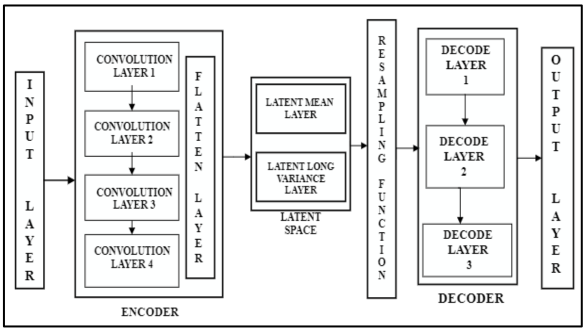

Fig. 4: Deep autoencoder architecture

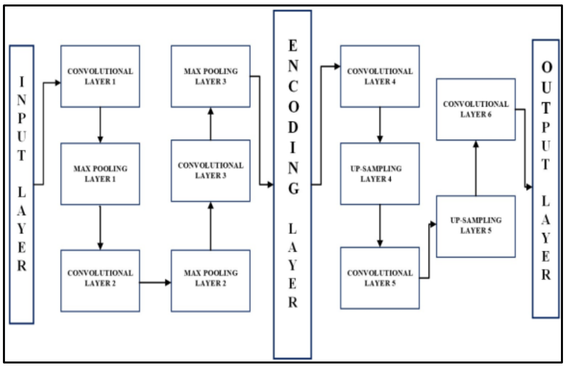

Fig. 5: Variational autoencoder architecture

Variational Autoencoder

Variational Autoencoders (VAEs) are a type of generative model that expands the functionalities of autoencoders with probabilistic modeling with a different approach to learning the latent representations compared to other models. Assuming a directed graphical model for the data generation procedure, they employed encoders which are the recognition models and decoders which are the generative models with parameters θ and Φ, respectively, to learn an approximation of the posterior distribution. The model is optimized using the training approach, which also includes a second loss component and the stochastic gradient variational bayes estimator. In contrast to regular autoencoders, variational autoencoders' probability distribution of the latent vector often resembles the training data. Variational autoencoders offer greater control over the modeling of the latent distribution. After training, sampling from the distribution and decoding can generate new data. During training, backpropagation is used to calculate the relationship of each network parameter with the final output loss. However, extra attention is required when sampling from the distribution. The distributions of the learned latent variables will be influenced by a *KL* divergence term in the loss function:

$$KL(q(z|x)||p(z)) = E[logq(z|x) - logp(z)] \tag{10}$$

The conventional definition of a VAE's probabilistic framework is defined below. To maximize the Evidence Lower Bound (ELBO), the VAE is given as:

$$ELBO = Eq(b|a)[\log(p(a|b))] - KL\ (q(b|a)||p(\mathrm{b})) \tag{11}$$

Here, latent variables: "*B*" is presumptively distributed previously $p(b)$. "a," is presumed to follow a probability distribution and is one of the observed variables. The joint distribution of the observable and latent variables is shown by the expressions $p(a|b)$ and $p(a, b) = p(a|b)\, p(b)$. The reconstruction loss function measures the difference between input data and the reconstructed image. The Mean Squared Error (MSE) is used for continuous data a binary cross entropy is used for binary data and the Kullback-Leibler divergence (*KL* divergence) is used for penalizing the divergence between the learned latent distribution and the previous distribution:

$$MSE(x, \hat{x}) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2 \tag{12}$$

Equation 12 is for continuous data and for binary data Eq. 13 can be used as follows:

$$KL((q|x)||p(z)) = -\frac{1}{2}\sum_{i=1}^{K}(1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2 \tag{13}$$

Here, μ and σ are the mean and standard deviation of the learned latent distribution $q(z|x)$ and $p(z)$ is the prior distribution (usually a standard Gaussian). The total loss function for a VAE is given as the sum of reconstruction loss and *KL* divergence as given in Eq. (14):

$$Loss = Reconstruction\ loss + KL\ Divergence \tag{14}$$

Figure 5 portrays the architecture of a variational autoencoder that uses probability distribution of the latent vector in addition to the convolutional function.

It also has a resampling function after which the decoder decodes the output image which is free of noise.

Convolutional Autoencoder

Convolutional autoencoders use convolutional layers to produce a compressed image format, by using the dimensionality reduction principle. This is an unsupervised learning model that compresses the image to reduce or remove the noise and keeps the useful and robust features intact. The use of convolutional features is the basic difference between the traditional autoencoders and convolutional ones. The encoder converts the input data to compressed latent space and the decoder reconstructs the compressed latent space representation to produce an output image that is remarkably close to the input but noise-free (Li *et al*., 2024). The block diagram of the Convolutional Autoencoder (CAE) is given in Fig. 6. It consists of an encoder convolutional network that converts the input to a latent space from which the decoder reconstructs the output by desampling the latent expression to noise-free data.

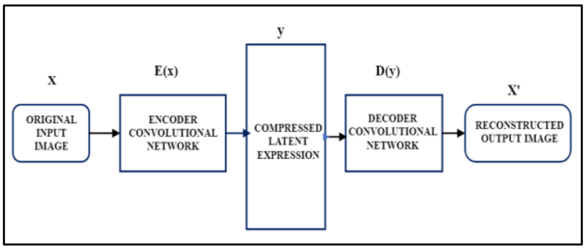

Fig. 6: Convolutional autoencoder architecture

Mathematically, the reconstruction error e^{CAE} is used to assess the convolutional autoencoder performance which is defined by:

$$e^{CAE} = L_{CAE}((\hat{a}^{(k)}, a^{(k)})) \tag{15}$$

L_{CAE} is the difference measurement which is the square Euclidean distance, which is given as given below:

$$L_{CAE}((\hat{a}^{(k)}, a^{(k)})) = \frac{1}{2} \parallel \hat{a}^{(k)} - a^{(k)} \parallel^2 \tag{16}$$

The cost function can therefore be expressed in its generic form as follows:

$$J_{CAE} = \frac{1}{M}\sum_{k=1}^{M} L_{CAE}(D(E(a^{(k)})), a^{(k)} \tag{17}$$

The ideal weight settings for the convolutional autoencoder are determined by minimizing the cost function J_{CAE}. Furthermore, C_{AEs} are often prone to overfitting, which implies that under some conditions, high-quality data may not be obtained for fitting a deep learning model. The level of compression is influenced by both the $C_{AE's}$ architecture and the encoder's output (Pintelas *et al.*, 2021).

Validation

It is an essential step in machine learning to ensure that the model has learned meaningful patterns from the data and is not simply memorizing the training data. In this study, validation for each of the five autoencoder models namely convolutional autoencoders, variational autoencoders, sparse autoencoders, deep autoencoders and denoising autoencoders to denoise the underwater images has been performed. To validate the models, the dataset has been split into 250 images for the training set and 150 images for the testing set. Each model has been trained using the training dataset and the testing set is used to estimate its performance. To determine the difference between the original and denoised images, the Peak Signal-to-Noise Ratio (PSNR).

Was employed to calculate the signal-to-noise ratio. Performance is enhanced when the PSNR is higher. The structural index of the images in the models was also examined using SSIM.

To ensure that the models were not overfitting to the training data, the PSNR values throughout the training process. If the model's performance on the testing set started to decrease while the performance on the training set continued to improve, the training process was paused. It is worth noting that PSNR is a commonly used metric for image denoising tasks but it does have its limitations. For example, it does not always correspond well with human perception of image quality and may not capture certain aspects of the image that are important for the task at hand. Hence SSIM was also used to analyse the result.

SSIM is a widely used image quality metric that measures the structural similarity between two images. A score of 1 indicates a complete resemblance between the two images. The SSIM index has a range of 0-1. The SSIM value can be used in the context of underwater image denoising to compare the denoised image with the original image and estimate the eminence of the denoised image. The model efficiency has been estimated on the validation set and the SSIM value can be used. The SSIM index can be calculated between the denoised image and the corresponding original image. If the SSIM value is higher it implies that the denoised image resembles the original image, which suggests that the model performed well. We also performed a visual inspection of the denoised images to ensure that the models were producing high-quality denoised images that were visually similar to the original images. By performing thorough validation, it is ensured that the autoencoder models were producing denoised images of high quality and were not overfitting to the training data.

Results

This study implemented five types of autoencoders (denoising, sparse, deep, convolutional and variational) on the EUVP dataset to compare their performance. The denoising autoencoder was used to remove noise, the sparse autoencoder to generate sparse representation, the deep autoencoder to learn multiple levels of representation and the variational autoencoder to generate a continuous and smooth latent space. The datasets varied in size, image quality and complexity. The study found that the performance of each autoencoder model varied depending on the dataset and the task at hand. The experiments performed on these datasets revealed that, based on the dataset and the task at hand, different autoencoders performed better or worse. For example, it is observed that the simple denoising autoencoder performed well on the MNIST dataset, but not as well on the CBIS-DDSM dataset. Similarly, here it is observed that the sparse autoencoder had lower SSIM and PSNR when compared to denoising autoencoders.

Results of Regulated Autoencoders

The original input and the noise-removed outputs are given below for the five different autoencoders (Figs. 9-13).

DOI: 10.3844/jcssp.2024.670.681

The original and denoised images obtained for the five autoencoders such as denoising autoencoder (Fig. 9), sparse autoencoder (Fig. 10), deep autoencoder (Fig. 11), variational autoencoder (Fig. 12), convolutional autoencoder (Fig. 13). From the experiments which have been conducted, we can understand and analyze the results of regularized autoencoders both qualitatively and quantitatively. Our results showed that the performance of each autoencoder varied with respect to the dataset and the application.

Table 1: PSNR and SSIM values for autoencoders on the EUVP dataset

	Metrics for image quality	
Type of autoencoder	Peak Signal to Noise Ratio (PSNR)	Structural Similarity Index (SSIM)
Denoising autoencoder	28.97	0.2530
Sparse autoencoder	28.83	0.2380
Deep autoencoder	28.82	0.2313
Variational autoencoder	28.86	0.2305
Convolutional autoencoder	28.46	0.2302

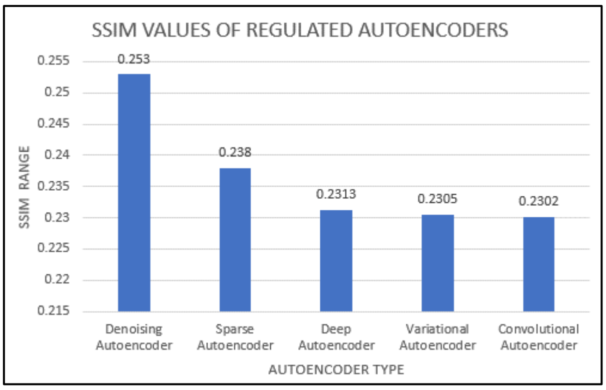

Fig. 7: Bar chart representation of structural similarity index

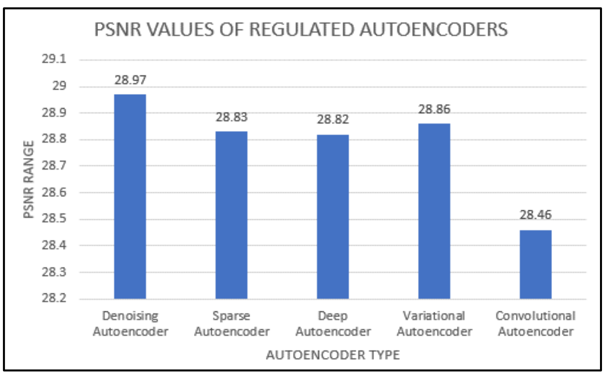

Fig. 8: Bar chart representation of peak signal-to-noise ratio

Figure 7 bar chart representation of structural similarity index autoencoder performed comparatively well on the EUVP dataset, but not as well on the MNIST dataset. The PSNR and SSIM values of the autoencoders are listed in Table 1.

The Figs. 7-8 shows the bar chart representation of the Structural Similarity Index measure (SSIM) and the Peak Signal to noise Ratio (PSNR) values obtained for the 5 autoencoders we have worked. From the graphical representation we can see that the denoising autoencoder has the highest SSIM and good PSNR values.

Outputs of Autoencoders

Fig. 9: Output of denoising autoencoder

Sparse Autoencoder

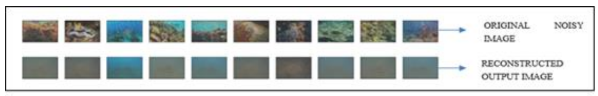

Fig. 10: Output of sparse autoencoder

Deep Autoencoder

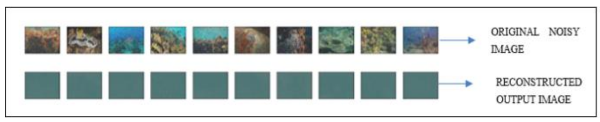

Fig. 11: Output of deep autoencoder

Variational Autoencoder

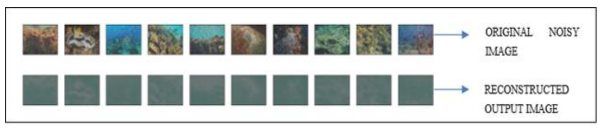

Fig. 12: Output of variational autoencoder

Convolutional Autoencoder

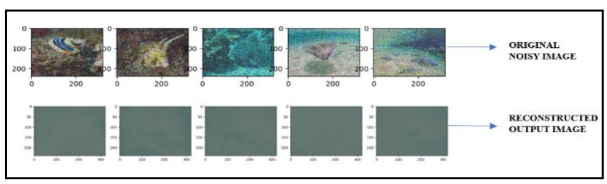

Fig. 13: Output of convolutional autoencoder

Quantitatively from the results obtained, it is observed that the Peak signal-to-noise Noise Ratio (PSNR) is high for denoising autoencoders and the Structural Similarity Index Measure (SSIM) is also best suited for denoising autoencoders in comparison with other autoencoders. DAE has achieved better performance among all the five autoencoders in terms of visual quality. The denoising autoencoder is the best suited for denoising of underwater images in terms of PSNR and SSIM. The results obtained by DAE when we have been seen with the eyes, it is clear that DAE performs better. So, it can be concluded that the denoising autoencoder is best suited for underwater images quantitatively and qualitatively. This experimental study provides insights into the performance of different autoencoder models and their applicability in underwater images.

Conclusion

Five different types of regulated autoencoder models were successfully implemented for the task of denoising underwater images in the EUVP dataset. The results show that all five models were able to significantly reduce the noise in the images and improve their quality, with the denoising autoencoder model performing slightly better than the others. The PSNR and SSIM metrics have been used to evaluate the performance of the models and ensured that they did not overfit the training data. Our work differs from pre-existing works on denoising underwater images in the use of a diverse dataset, a systematic evaluation approach and the comparison of multiple autoencoder models. Despite some limitations, such as the limited availability of underwater image datasets and the need for significant computational resources, our approach has shown promising results in denoising underwater images.

The sparse autoencoder has been tuned in a manner that it could work for underwater images. Modifications were applied by adding a drop-out layer after each encoder layer. By doing so the issue of overfitting has been effectively overcome and its ability to generalize to unseen data was improved. Thus, the efficiency and robustness of the Sparse autoencoder were improvised using our model.

Future Works

In terms of future work, it is better to explore the use of transfer learning to further improve the performance of these models. Pre-trained models also can be applied to large datasets and can be fine-tuned on smaller underwater image datasets. Additionally, we can investigate the use of generative models such as GANs to enhance underwater images. The dataset can be expanded to include more diverse types of underwater images and noises to further test the robustness of the models. Overall, our work shows that denoising models can be efficient and effective providing a starting point for further exploration of this subject area.

Funding Information

The authors have not received any financial support or funding to report.

Author's Contributions

Shreya Priyadarshini Roy: Conceptualization, conducted all the experiments and investigated the data analysis.

Sivagami M.: Organized the study reviewed the results and checked the experiments done, validated and funded acquisition.

Angelin Beulah S.: Reviewed and revised the data analysis and contributed to written the manuscript.

Maheswari N.: Reviewed and funded acquisition.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all the other authors have read and approved the manuscript and that no ethical issues are involved.

Conflicts of Interest

The authors declare that they do not have any conflicts of interest.

References

Arbahri, D., Nurhayati, O. D. & Mudita, I. (2024). 'Machine Learning Oceanographic Data for Prediction of the Potential of Marine Resources.' *Journal of Computer Science*, *20*(2),129-139. https://doi.org/10.3844/jcssp.2024.129.139

Bank, D., Koenigstein, N., & Giryes, R. (2023). Autoencoders. *Machine Learning for Data Science Handbook: Data mining and Knowledge Discovery Handbook*, 353-374. https://doi.org/10.1007/978-3-031-24628-9_16

Chandra, I. S., Shastri, R. K., Kavitha, D., Kumar, K. R., Manochitra, S., & Babu, P. B. (2023). CNN based color balancing and denoising technique for underwater images: CNN-CBDT. *Measurement: Sensors*, *28*, 100835. https://doi.org/10.1016/j.measen.2023.100835

Cherian, A. K., Poovammal, E., Philip, N. S., Ramana, K., Singh, S., & Ra, I. H. (2021). Deep learning based filtering algorithm for noise removal in underwater images. *Water*, *13*(19), 2742. https://doi.org/10.3390/w13192742

Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007). Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, *16*(8), 2080-2095. https://doi: 10.1109/TIP.2007.901238

Du Toit, T., Kruger, H., & Van Der Merwe, A. (2023). Automated terrain classification with a Bayesian hyperparameter optimized deep supervised autoencoder model. https://doi.org/10.3844/jcssp.2023.1073.1086

Fan, L., Zhang, F., Fan, H., & Zhang, C. (2019). Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine and Art*, *2*(1), 7. https://doi.org/10.1186/s42492-019-0016-7

Gondara, L. (2016, December). Medical image denoising using convolutional denoising autoencoders. *In 2016 IEEE 16th International Conference on Data Mining Workshops* (*ICDMW*), (pp. 241-246). IEEE. https://doi.org/10.1109/ICDMW.2016.0041

Goyal, B., Dogra, A., Agrawal, S., Sohi, B. S., & Sharma, A. (2020). Image denoising review: From classical to state-of-the-art approaches. *Information Fusion*, *55*, 220-244. https://doi.org/10.1016/j.inffus.2019.09.003

Han, M., Lyu, Z., Qiu, T., & Xu, M. (2018). A review on intelligence dehazing and color restoration for underwater images. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, *50*(5), 1820-1832. https://doi.org/10.1109/TSMC.2017.2788902

Hashisho, Y., Albadawi, M., Krause, T., & von Lukas, U. F. (2019, September). Underwater color restoration using u-net denoising autoencoder. In 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA), (117-122). *IEEE*. https://ieeexplore.ieee.org/abstract/document/8868679

Hou, G. J., Luan, X., & Song, D. L. (2015, June). A study on color model selection for underwater color image preprocessing. *In 2015 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems* (*CYBER*), (1456-1461). *IEEE*. https://ieeexplore.ieee.org/abstract/document/7288159

Jeelani, A., & Veena, M. B. (2017). Denoising the underwater images by using adaptive filters. *European Journal of Applied Sciences*, *5*(2), 01-01. https://doi.org/10.14738/aivp.52.2939

Iqbal, K., Salam, R. A., Osman, A., & Talib, A. Z. (2007). Underwater Image Enhancement Using an Integrated Colour Model. *IAENG International Journal of Computer Science*, *34*(2). https://doi.org/10.1049/iet- ipr.2016.1075

Kalantari, N. K., Bako, S., & Sen, P. (2015). A machine learning approach for filtering Monte Carlo noise. *ACM Trans. Graph.*, *34*(4), 122-1. https://doi.org/10.1145/2766977

Li, Y., Li, D., Gao, Z., Wang, S., & Jiao, Q. (2024). Underwater image enhancement utilizing adaptive color correction and model conversion for dehazing. *Optics and Laser Technology*, *169*, 110039. https://doi.org/10.1016/j.optlastec.2023.110039

Pintelas, E., Livieris, I. E., & Pintelas, P. E. (2021). A convolutional autoencoder topology for classification in high-dimensional noisy image datasets. *Sensors*, *21*(22), 7731. https://doi.org/10.3390/s21227731

Skribtsov, P. V., & Surikov, S. O. (2016). Regularization method for solving denoising and inpainting task using stacked sparse denoising autoencoders. *American Journal of Applied Sciences*, *13*(1), 64-72. https://doi.org/10.3844/ajassp.2016.64.72

Thakur, R. S., Chatterjee, S., Yadav, R. N., & Gupta, L. (2021). Image de-noising with machine learning: A review. *IEEE Access*, *9*, 93338-93363. https://doi.org/10.1109/ACCESS.2021.3092425

Xiang, Q., & Pang, X. (2018, October)., 'Improved denoising auto-encoders for image denoising.' *In 2018 11th International Congress on Image and Signal Processing*, *Bio Medical Engineering and Informatics (CISP-BMEI)*, (*1*-9). IEEE. https://doi.org/10.1109/CISP-BMEI.2018.8633143

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., ... & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI open*, *1*, 57-81. https://doi.org/10.1016/.aiopen.2021.01.001