International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 16 (2024) pp. 535–546

Article

Double-U Net: A Novel Approach for Deep Learning Based Image Defogging

Anu Bajaj [1,*], Ankit Bhardwaj [1], Yessica Tuteja [1], Mayank Jindal [1], Surbhi [1], Rahul Saini [1] and Ajith Abraham [2]

[1] Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala 147004, Punjab, India; ankitbhardwaj1743@gmail.com (A.B.); yessicatuteja@gmail.com (Y.T.); mkjindal113@gmail.com (M.J.); khoslasurbhi14@gmail.com (S.); rahulsaini.2k2@gmail.com (R.S.)
[2] School of AI, Bennett University, Greater Noida 201310, Uttar Pradesh, India; ajith.abraham@ieee.org
* Correspondence author: er.anubajaj@gmail.com

Received date: 14 February 2024; Accepted date: 24 April 2024; Published online: 10 July 2024

Abstract: Image defogging has become a major difficulty in the quickly developing field of digital photography. Addressing this issue is crucial, given the growing need for crisp, high-quality images in industries like social networking, entertainment, navigation, and surveillance. Researchers from all over the world have put out a variety of presumptions and techniques to improve clarity in hazy or foggy images. This research paper proposed a new deep learning algorithm, the Double U-Net algorithm. Which is a concatenation of two similar or distinct U Net architectures to produce the best possible outcomes while enhancing visibility in cloudy photos. We conduct a thorough analysis to compare the effectiveness of the proposed algorithm with other state-of-the-art defogging methods, considering factors such as robustness to varying fog intensities and image features, computing efficiency, and visual quality. It is observed that the proposed architecture outperformed other techniques in terms of PSNR (26.88) and SSIM (0.99979). The findings demonstrate that the proposed algorithm performs exceptionally well in improving visibility and recovering fine-grained image information under various atmospheric situations.

Keywords: image processing; double U-Net architecture; image defogging; image dehazing; CNN; autoencoder; U-Net; PSNR; SSIM; accuracy; BCE

1. Introduction

Image processing has evolved rapidly in recent years, with various techniques developed to improve image quality [1]. In this field, one of the crucial challenges is dealing with degraded visibility and blurred appearance caused by environmental factors such as fog, haze, or other atmospheric conditions [2]. These factors can significantly affect the quality and usability of images [3]. To address this challenge, an image-defogging technique is introduced to enhance the visibility of images degraded by fog or haze [4]. It involves removing the effects of atmospheric scattering, which causes the image to appear hazy or blurry. There are several types of image-defogging techniques, including single-image and multi-image defogging methods. Single-image defogging techniques use only one image to remove the effects of haze, while multi-image defogging techniques use multiple images to improve the image quality [5].

Various image-defogging techniques have been developed for solving real-world problems like transportation that improve the visibility of images captured by cameras on vehicles, thereby enhancing driver safety [6]. It is also helpful in surveillance, making identifying suspects easier [7]. However, image-defogging techniques also have limitations. One of the major shortcomings is the computational complexity of the methods, which can make them impractical for real-time applications [8]. Another limitation is the accuracy of the techniques, which can be affected by factors such as lighting conditions and the type of haze [9]. Researchers developed deep learning algorithms to overcome these limitations for more efficient and accurate image-defogging techniques. These algorithms improve the performance

by identifying the intricate patterns of the image. Architectures like CNN [10], Autoencoders [11], and U Net [12] are state-of-the-art deep learning tools used to solve many real-world problems [13].

CNNs were among the first image-processing deep-learning models to be implemented practically in the last decade. These models used the 'convolutional approach' to localize essential features in an image array to identify them [14]. These models kick-started computer vision research and became the foundation for more advanced Deep Learning models and architectures, including the Autoencoders and U-Nets [15]. Autoencoders have an encoder-decoder architecture, which allows them to shrink original images to a minimum and reconstruct them into the desired output, making them suitable for image denoising and recolorization tasks [16]. U-Nets are one step ahead of autoencoders, with concatenated skip connections, allowing them to perform higher-level image transformation tasks such as dehazing and, to some extent, style transfer [17].

In this paper, we have proposed an image defogging architecture, i.e., double-U Net Architecture, which is a combination of image fusion and dehazing techniques to improve image quality. This architecture works by estimating the haze transmission map, which represents the amount of haze in each pixel of the image. Double U Net architecture applies a haze removal process to recover clear and sharp images using the estimated transmission map. Some key advantages of the Double-U Net architecture include its ability to handle different types and levels of haze, its effectiveness in preserving image details and colors, and its computational efficiency. With its ability to effectively remove haze, preserve image details and colors, and its computational efficiency, the Double-U Net architecture is a powerful tool for enhancing visibility and improving the quality of foggy or hazy images.

The organization of this paper is as follows: Section 2 briefs the deep learning algorithms used for image defogging. The working of the baseline models and the proposed algorithm is presented in Sections 3 and 4. Section 5 describes the experimental setup, followed by the results and discussions in Section 5. The paper is concluded in Section 6.

2. Related Works

Tufail et al. [18] proposed a novel approach to enhance image-defogging techniques by optimizing the transmission map, particularly leveraging the dark channel before estimating atmospheric light more accurately. Their methodology involved adaptive selection of the transmission map, determined by the fog density within the image. Additionally, they utilized Laplacian and guided filters to refine the transmission map, significantly improving the visibility of images, especially those with expansive sky regions. Deshmukh et al. [19] addressed the major issue of fog-related accidents for vehicles. Their proposed method included a complete embedded system for restoring foggy images, utilizing the Mean Channel Prior (MCP) algorithm for defogging. The system employed a Raspberry Pi, camera, and display screen, making it a low-power, portable, and standalone system. They compared the results based on Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR), concluding its superiority, especially in scenarios with varying levels of fog density.

Wang et al. [20] presented a linear model to estimate the transmission function efficiently. They employed a quadtree to search for a region that best represents the scatter of airlight, demonstrating comparable results to state-of-the-art techniques while significantly improving efficiency. Mao et al. [21] introduced a novel method for defogging single images using multi-exposure image fusion and detail enhancement. Their efficient algorithm fused multiple images with different exposures to enhance defogged image visibility. The method also included a detail enhancement technique, which further improved the quality of the defogged image. Kumari and Sahoo [22] presented a novel visibility restoration approach for remote sensing images affected by adverse weather conditions like haze and fog. The method was based on segmentation and unsharp mask-guided filtering, first employing a segmentation method to determine atmospheric light and then estimating the transmission map using the dark channel prior. To address halo artifacts and inconsistencies in the resultant image, they utilized a guided filter method based on unsharp masking to optimize the transmission map. Their experimental results demonstrated high uniformity in qualitative and quantitative evaluations using six performance metrics.

Lieu et al. [23] conducted a comprehensive study addressing significant challenges in image defogging. They introduced the Multiple Real-World Foggy Image Dataset (MRFID), containing foggy and clear images from 200 outdoor scenes. Processing these images using 16 defogging methods, they comprehensively evaluated the visual quality of the defogged images. Additionally, they developed a new Fog-relevant Feature-based Similarity index (FRFSIM) for assessing the visual quality of defogged images, showing promising results in their extensive experimental evaluations. This work significantly contributes to the literature by addressing the lack of real-world foggy image datasets and introducing a new image quality assessment method.

Tannistha et al. [24] presented a significant contribution to fog removal from images, addressing two critical challenges: the lack of real-world foggy image datasets and the absence of suitable image quality assessment methods. They introduced the SAMEER–TU database, containing 5,390 images captured under varying visibility conditions. Additionally, they proposed a method for enhancing the visibility of foggy images. Qu et al. [25] conducted a study on defogging optical remote sensing images, introducing a new algorithm that adjusts the transmittance attenuation values across three channels based on different wavelengths. Their method incorporated deep learning techniques for segmenting dense fog areas and estimating global atmospheric light, showing promising results in improving image clarity and quality. Anan et al. [26] introduced a novel framework for defogging foggy images, addressing the limitations of the Dark Channel Prior (DCP) technique. The approach involved segmenting images into sky and non-sky regions, restoring these parts separately, and blending them. By using Contrast Limited Adaptive Histogram Equalization (CLAHE) for the sky and a modified DCP for the non-sky regions, the proposed method achieved improved visual quality and lower processing time than other techniques.

Hassan et al. [27] proposed a cascade strategy for foggy image defogging, combining CLAHE and the No-Black Pixel Constraint with Planar Assumption (NBPC + PA). The proposed algorithm achieved better defogging results for homogeneous and inhomogeneous fog by resolving CLAHE's limitations and optimizing parameters. Chen et al. [28] proposed an integrated image-defogging network that combines an improved atmospheric scattering model with attention feature fusion. The approach addressed uneven fog density in images by leveraging an attention mechanism. The proposed method achieved robust defogging results by treating thick fog and mist differently. However, when applied to real natural foggy images, some patches may appear in the restored images. Overall, the approach reasonably removes thick fog and mist while preserving image quality.

3. Existing Models

The most widely used base models for general defogging and image transformation tasks are the Convolutional Neural Network (CNN), Autoencoders, and the U-Net. Each of these models is explained one by one in this section

3.1. Convolutional Neural Networks (CNN)

A Convolutional Neural Network (CNN) is a deep learning architecture designed for processing unstructured data, most commonly images. CNNs are masters in identifying localized patterns within the pixel values of images, making them quite suitable for tasks like image classification and segmentation and, in some cases, image transformation. It has convolutional layers that learn high and low-level hierarchical features by itself. The proposed Convolutional Network differs from conventional CNNs, as it does not include pooling or deep hidden layers. It consists of Convolutional layers, with a uniform kernel size of (1,1), to ensure consistency in input and output shape. This is done since the task is not image classification but rather image transformation from densely foggy images to non-foggy images, thus making this CNN architecture somewhat border-line Autoencoder. The hyperparameters, including the number of layers, filter size, kernel shapes, activation functions, and size of layers, are given in Table 1.

Table 1. CNN Architecture.

Layers	Con0: 16 filters (1 × 1 size) Con1: 32 filters (1 × 1 size) Con2: 64 filters (1 × 1 size) Con3: 128 filters (1 × 1 size) Con4: 256 filters (1 × 1 size) Con5: 128 filters (1 × 1 size) Con6: 64 filters (1 × 1 size) Con7: 32 filters (1 × 1 size) Con8: 16 filters (1 × 1 size) Activation functions: RelU Con9: 3 filters (1 × 1 size) Activation function: Sigmoid
Input Shape	128 × 256 × 3
Output Shape	128 × 256 × 3
Optimization	Adam
Loss Function	Binary Cross Entropy
Performance Matrix	Accuracy
Epochs	20

3.2. Autoencoder

An autoencoder is a deep learning architecture for unsupervised learning and feature extraction from image datasets. It usually consists of two parts: an encoder, employing many convolutional layers for spatial feature extraction, and a decoder, using similar convolutional layers for image reconstruction, forming the output. The encoder section of this architecture downsampled the image to the bottom layer, while the decoder section resamples it back to the shape of the output, thus giving this architecture a double-sided funnel-like shape. The Autoencoder developed for this research consists of three convolutional layers with strides in the encoder section, whereas convolutional layers are along with three upsampling layers in the decoder section. The detailed structure and respective hyperparameters are explained in Table 2.

Table 2. Autoencoder Architecture.

Convolutional & Upsampling Layers	Con0: 16 filters (3 × 3 size, 2 strides) Con1: 32 filters (3 × 3 size) Con2: 64 filters (3 × 3 size, 2 strides) Con3: 128 filters (3 × 3 size) Con4: 256 filters (3 × 3 size, 2 strides) Con5: 256 filters (3 × 3 size) Con6: 256 filters (3 × 3 size) UpSamp0: (2 × 2) Con7: 128 filters (3 × 3 size) Con8: 64 filters (3 × 3 size) UpSamp1: (2 × 2) Con9: 32 filters (3 × 3 size) Con10: 16 filters (3 × 3 size) UpSamp2: (2 × 2) Activation functions: RelU Con11: 3 filters (1 × 1 size) Activation function: Sigmoid
Input Shape	128 × 256 × 3
Output Shape	128 × 256 × 3
Optimization	Adam
Loss Function	Binary Cross Entropy
Performance Matrix	Accuracy
Epochs	20

3.3. U-Net Architecture

U-Net is a robust deep-learning architecture widely employed for image transformation and denoising tasks. Its distinctive feature is its recognizable U-shaped architecture, consisting of a contracting path for capturing contextual information and an expansive path for precise localization. A U-Net can be considered a specialized, upgraded Autoencoder since it shares its core structure with an Autoencoder architecture, i.e., it comprises an encoder, base, and decoder. The key difference lies in the intermediate layers on encoders and decoders sharing the same output shape, which are concatenated with the help of skip connections, as shown in Figure 1.

The encoder part comprises convolutional and pooling layers, effectively extracting hierarchical features, while the decoder, symmetrically connected through skip connections, reconstructs the segmented output with fine-grained details. U-Net's ability to integrate global and local features, facilitated by skip connections, makes it particularly effective for tasks requiring pixel-wise transformation, such as defogging/dehazing in our case. The architecture’s versatility and robustness have extended its applications to various domains where accurate image reconstruction and transformation are essential. Table 3 provides the U-Net architecture.

Table 3. U-Net description.

Convolutional & TansConvo Layers	2 × Con0: 16 filters (3 × 3 size) Maxpooling(2,2) 2 × Con1: 32 filters (3 × 3 size) Maxpooling(2,2) 2 × Con2: 64 filters (3 × 3 size) Maxpooling(2,2) 2 × Con3: 128 filters (3 × 3 size) Maxpooling(2,2) 2 × Con4: 256 filters (3 × 3 size) Maxpooling(2,2) 2 × TransCon0: 128 filters (3 × 3 size) strides(2,2) 2 × TransCon1: 64 filters (3 × 3 size) strides(2,2) 2 × TransCon2: 32 filters (3 × 3 size) strides(2,2) 2 × TransCon3: 16 filters (3 × 3 size) strides(2,2) Activation function: Sigmoid
Input Shape	128 × 256 × 3
Output Shape	128 × 256 × 3
Optimization	Adam
Loss Function	Binary Cross Entropy
Performance Matrix	Accuracy
Epochs	20

4. Proposed Model (Double U Net Algorithm)

A novel neural network model, the Double-U Net algorithm, is proposed to perform the required defogging task. At its core, Double-U Net is a concatenation of two identical or distinct U-Net models. In other words, the Double-U Net model concatenates two U-Net architectures sequentially placed one after the other. Therefore, the output produced by the first U-Net becomes the input of the subsequent U-Net. This sequential architecture, as per our observations, provides multiple benefits. The Sequential Nature of the Architecture allows hierarchical learning, which is extremely useful in dehazing/defogging tasks as the first U-Net cycle captures lower-level features, and the next cycle refines them into higher-level features. This architecture also allows progressive refinement of images, making image transformation/denoising more efficient and accurate. Furthermore, it provides immense flexibility to the architecture, as each U-Net cycle can be tweaked and modified separately without disturbing the consistency of input and output layers. An overview of the architecture of Double-U Net and its working is briefly explained using the diagram in Figure 1.

The number of layers and nodes on each layer of both the U-Nets depend entirely on the magnitude of the dataset to be trained and the dimensions of the image. After weeks of experimentation and testing various variations of this architecture on multiple sun sections of the foggy Cityscapes dataset [24], we came up with the idea of using a distinct pair of U-Net for concatenation, with the later (second) U section having double the nodes at each layer than the former (first) U section. However, the number of hidden layers remains the same for both sections. This allowed the architecture to capture low-level features in the first section and higher-level features in the subsequent section. Table 4 exhibits the details of the structure and hyperparameters used to make the Double U-Net architecture.

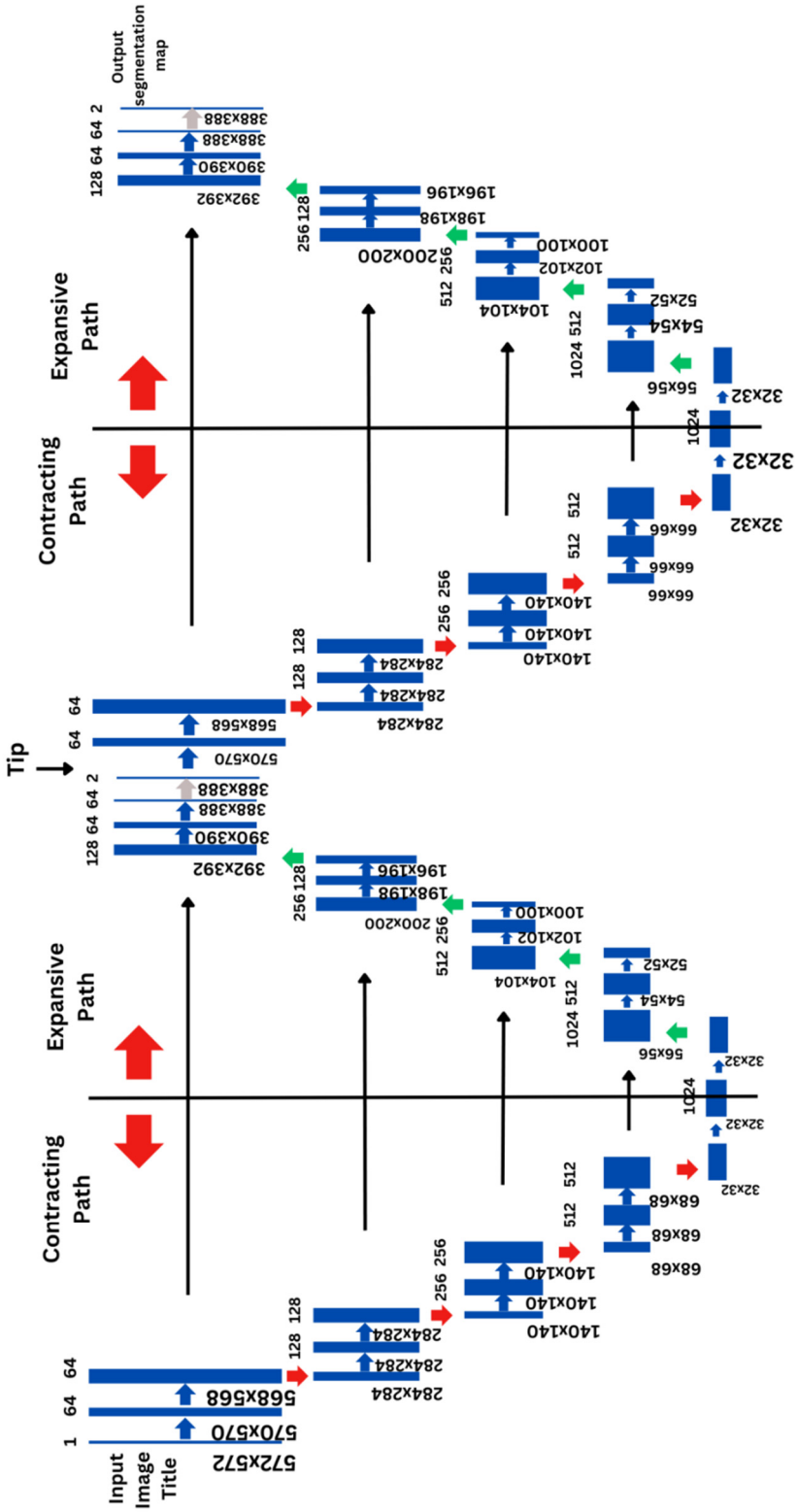

Figure 1. The Double-U Net Architecture.

Table 4. The Double-U Net Architecture.

Convolutional & TansConvo Layers	2 × Con0: 16 filters (3 × 3 size) Maxpooling(2,2) 2 × Con1: 32 filters (3 × 3 size) Maxpooling(2,2) 2 × Con2: 64 filters (3 × 3 size) Maxpooling(2,2) 2 × Con3: 128 filters (3 × 3 size) Maxpooling(2,2) 2 × Con4: 256 filters (3 × 3 size) Maxpooling(2,2) 2 × TransCon0: 128 filters (3 × 3 size) strides(2,2) 2 × TransCon1: 64 filters (3 × 3 size) strides(2,2) 2 × TransCon2: 32 filters (3 × 3 size) strides(2,2) 2 × TransCon3: 16 filters (3 × 3 size) strides(2,2) 2 × Con5: 16 filters (3 × 3 size) Maxpooling(2,2) 2 × Con6: 32 filters (3 × 3 size) Maxpooling(2,2) 2 × Con7: 64 filters (3 × 3 size) Maxpooling(2,2) 2xCon8: 128 filters (3 × 3 size) Maxpooling(2,2) 2 × Con9: 256 filters (3 × 3 size) Maxpooling(2,2) 2 × TransCon4: 128 filters (3 × 3 size) strides(2,2) 2 × TransCon5: 64 filters (3 × 3 size) strides(2,2) 2 × TransCon6: 32 filters (3 × 3 size) strides(2,2) 2 × TransCon7: 16 filters (3 × 3 size) strides(2,2) Activation function: Sigmoid
Input Shape	128 × 256 × 3
Output Shape	128 × 256 × 3
Optimization	Adam
Loss Function	Binary Cross Entropy
Performance Matrix	Accuracy
Epochs	20

5. Experimental Setup

The experiment is performed by training the model on an NVIDIA TESLA P100 cloud GPU. The models were created, trained and deployed on Kaggle [29], a widely used online Data Science platform. To work with Neural Network Architectures, we have used the TensorFlow library, as well as a few other supporting libraries like Numpy, Pandas, OpenCV2, and Matplotlib. The dataset used for this research is Foggy Cityscapes, a Benchmark dataset that is a derivative of the widely used Cityscapes dataset [30]. This is a synthetic foggy dataset that simulates fog in real scenes. Each foggy image is rendered with a clear image and depth map from Cityscapes. Thus, the annotations and data split in Foggy Cityscapes are inherited from Cityscapes. The dataset contains a total of 10,425 images of size 1,024 × 2,048 pixels. Each image in the dataset has three alpha channel values, 0.005, 0.01, and 0.02, resulting in 3,475 distinct pictures with these three alpha channel values. To make the dataset fit for the proposed model, the images are scaled to 128 × 256 pixels and divided into three categories based on their alpha channel values:

1. 0.005:- No fog images
2. 0.01:- Medium foggy images

3. 0.02:- Dense foggy images

After experimentation with all the available classes, it was decided to drop the medium fog class altogether and train the model to transform the densely foggy images into non-foggy images directly. Furthermore, the RGB channel values have been scaled down from the range of 0–255 to 0–1 with the help of normalization. Next, we divide the dataset into a training set (3,000 images per class) and a testing set (475 images per class).

5.1. Loss Function: Binary Cross Entropy

A loss function is the most critical component in the training process of any neural network architecture, as it guides the optimization algorithm to update the model's parameters to minimize the distance of predicted values from the actual values. In this paper, we have used Binary Cross Entropy (BCE), also known as logarithmic loss or log loss, which is a loss function model metric that tracks incorrect labeling of the data class by a deep learning model. It is usually used to measure the performance of a classification model whose output is a probability value in the range of 0 to 1 [31]. Low log loss values equate to high accuracy values. Mathematically, it can be expressed as:

$$BCE\ (y\ ,\ y')\ =\ -\ (\ y\ .\ \log\ (\ \ y'\ \ )\ +\ (\ 1\ -\ y\ )\ .\ \log\ (\ 1\ -\ \ y'\ \ )\ ) \tag{1}$$

where y is the actual binary label (0 or 1) of the class, and y' is the predicted probability of a data point belonging to class 1.

5.2. Performance Metrics

To evaluate the performance of image-defogging techniques, we need to establish criteria for evaluation and performance metrics. The criteria for performance evaluation include the accuracy of the method in removing haze, the computational complexity of the method, and the robustness of the method to different types of haze. The performance metrics include the accuracy, peak signal-to-noise ratio (PSNR), and the structural similarity index (SSIM) [32], as discussed below:

i. Accuracy

Accuracy is the ratio of correctly classified points (prediction) to the total number of predictions. Its value ranges between 0 and 1 [33]. In simple terms, it is a measure of the correctness of predicted values.

$$Accuracy\ =\ \frac{number\ of\ correct\ predictions}{total\ number\ of\ predictions} \tag{2}$$

ii. Structural Similarity Index

The Structural Similarity Index (SSIM) is a perceptual metric that quantifies image quality degradation caused by processing such as data compression or by losses in data transmission. It is a complete reference metric that requires two images from the same image capture—a reference image and a processed image [34].

The SSIM index is calculated on various windows of an image. The measure between two windows x and y of common size N × N is:

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

Here, μ_x, μ_y are the averages of x and y and σ_x^2, σ_y^2 are the variances of x and y whereas σ_{xy} represents the covariance and c_1 and c_2 are two variables used to stabilize the division with weak denominator:

$$c_1\ =\ (k_1L)^2\ ,c_2\ =\ (k_2L)^2 \tag{4}$$

Here L is the dynamic range of pixel values, k_1 and k_2 are set to 0.01 and 0.03.

iii. Peak Signal-to-Noise Ratio (PSNR)

The Peak Signal-to-Noise Ratio (PSNR) is a measure used in image and video quality assessment. It is defined via the Mean Squared Error (MSE), which measures the average of the squares of the errors between a noise-free image and its noisy approximation [35].

Here's the formula for PSNR:

$$PSNR\ =\ 10log_{10}(\frac{(L-1)^2}{MSE})\ =\ 20log_{10}(\frac{L-1}{RMSE}) \tag{5}$$

Here, L is the maximum number of possible intensity levels and MSE is the mean square error:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(O(i,j) - D(i,j))^2 \quad (6)$$

Here, *O* and *D* are the original and degraded images.

6. Experimental Results and Analysis

This section gives insights into the performance of the proposed model based on model training analysis and the evaluation metrics as given below:

6.1. Model Training Analysis

The proposed Double-U Net model was trained on the training dataset of 3,000 images for 20 epochs. The average time to train the model per epoch was approximately 35 seconds, courtesy of the TESLA P100 Cloud GPU, and the entire training process took around 12 minutes. The model training started off well, with the first epoch yielding an impressive training accuracy of 0.8005 and loss of just 0.6060, which testifies the robustness and feature extraction power of this architecture. after 20 epochs, the architecture had an accuracy of 0.9225 and a training loss of just 0.5879, making it a highly reliable model. The graph in Figure 2 represents the training history of this model, which clearly indicates how the model was rapid and efficient in generalizing the datasets.

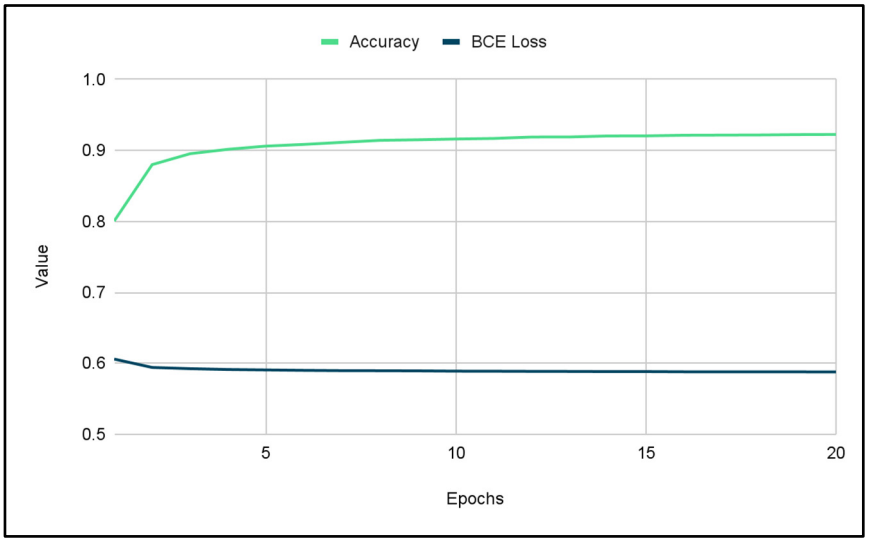

Figure 2. Training Graph of Double-U Net.

Figure 3 clearly shows the inconsistent fluctuations in the plot of compared models with that of the proposed model, which strongly indicates that these models are finding it hard to generalize well on the training data.

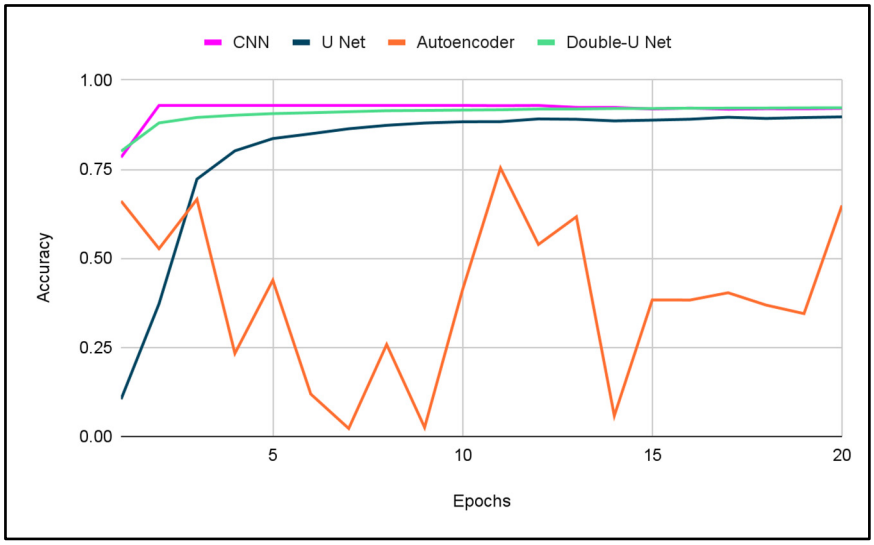

Figure 3. Training Accuracy Comparison of models.

6.2. Model Performance Analysis

The testing phase of the Double-U Net architecture solidified the findings from the training phase. The model was tested on a set of 475 images, and, upon evaluation, yielded an impressive accuracy of 0.9176 (91.78%) with a loss of just 0.5736. The output was then rescaled and converted back to the image dataset, which was further tested against the ground truth data to go through further evaluation metrics, that is, SSIM and PSNR. the mean Structural Similarity Index of the images produced by our model with the ground truth was an impressive 0.99978, which means that the structural similarity of the predicted and actual (ground truth) images is nearly similar. The PSNR value for the predicted dataset with respect to the original dataset was respectable at 26.88, a value that indicates good reconstruction. Upon comparing these results with the ones obtained from state of the art models, the consistency and robustness of Double-U Net become evident. As shown in Table 5, the Double-U Net yields superior results than all of its contemporary competitors, namely CNN, Autoencoder, and U-Net in every single performance metric we calculated.

Table 5. Performance Comparison of all the Models on Test Data.

Model Name	Loss	Accuracy	SSIM	PSNR
Double-U Net	**0.5736**	**91.76%**	**0.9997**	**26.88**
U-Net	0.5829	88.54%	0.9967	24.38
Autoencoder	111,347	14.26%	0.9822	07.71
CNN	0.5784	87.94%	0.9953	24.04

Finally, looking at the output result itself will help in truly absorbing the efficiency and validity of the Double-U Net. Figure 4 showcases one of the outputs produced by the Double-U Net architecture along with the original foggy image and the ground truth.

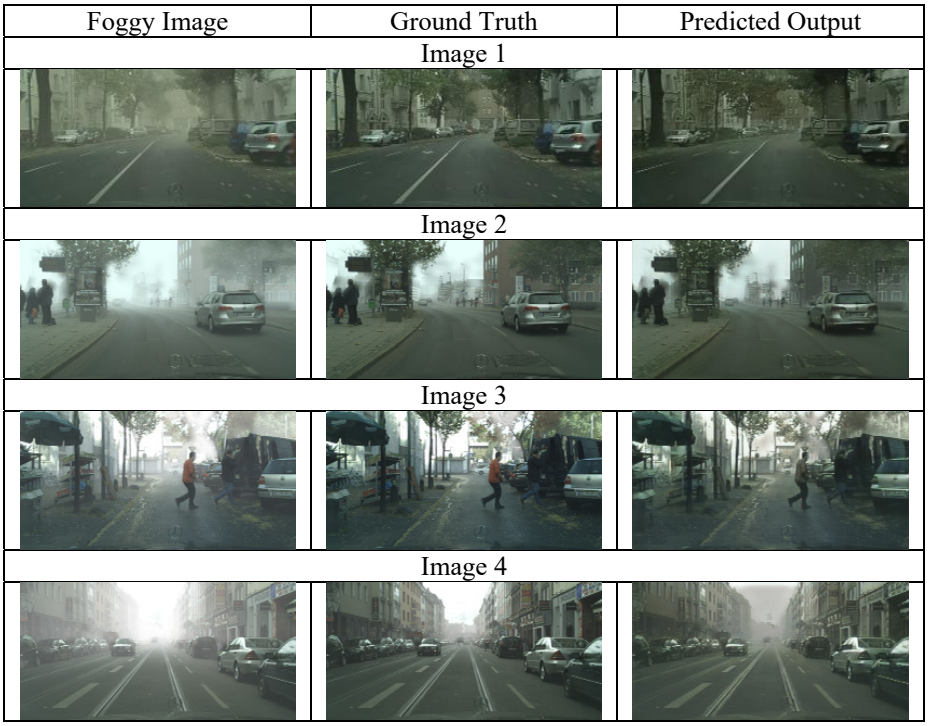

Figure 4. Comparison of predicted output with the ground truth.

7. Conclusions

In conclusion, we have proposed a improved version of U net architecture which is a fusion of two U Net architectures. The second U Net gets thes the input from the output of the first U Net thereby enhancing visibility in foggy or hazy images, addressing critical limitations in existing methods. The proposed algorithm exhibits superior performance in PSNR and SSIM compared to CNN, Autoencoder and U Net algorithms. In future we will apply the proposed algorithm on real-time quality enhancement which can be used for autonomous driving, surveillance systems, and remote sensing applications. Additionally, we will improve the performance of the proposed algorithm by using parameter tuning, and multi-modal fusion strategies.

Author Contributions

Conceptualization and Methodology: A.B. (Anu Bajaj); Project Administration: A.B. (Anu Bajaj) and Y.T.; Software Development: A.B. (Ankit Bhardwaj) and M.J.; Resources: R.S.; Data Curation: S.; Supervision: A.A. All authors have read and agreed to the published version of the manuscript.

Funding

The research received no external funding.

Conflict of Interest Statement

Authors declare no conflict of interest.

Data Availability Statement

Foggy Cityscapes dataset: https://paperswithcode.com/dataset/foggy-cityscapes.

References

1. A. Bajaj, T. Sharma, and O. Prakash Sangwan. 2020. “Information Retrieval in Conjunction With Deep Learning.” In *Advances in Computational Intelligence and Robotics Book Series*, 300–311.
2. P.S. Likhitaa and A. R, "A Comparative Analysis of Image Dehazing using Image Processing and Deep Learning Techniques," 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2021, pp. 1611-1616.
3. C. Yani, Z. Shuaiqing, L. Wenjin, D. Jiaxian and R. Jia “An improved dark channel defogging algorithm based on the HSI colour space.” *IET Image Processing* 16, no. 3 (2022): 823-838.
4. K. Pushpa and S. Kumawat. “A Review on Comparison of Different Techniques of Image Dehazing.” In *Proceedings of the 2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, pp. 1-6. IEEE, 2020
5. Z. Lin, Q. Luo, and Y. Jiang, et al. Image defogging based on multi-input and multi-scale UNet. *SIViP 17*, 1143–1151 (2023).
6. I. Ogunrinde and S. Bernadin, "A Review of the Impacts of Defogging on Deep Learning-Based Object

Detectors in Self-Driving Cars," In *Proceedings of the SoutheastCon 2021, Atlanta, GA, USA*, 2021; pp. 01-08.
7. T. Ershang and J. Kim. "De-hazing CCTV Images using Dark Channel Prior for Improved Vehicle Detection." In *Proceedings of the 2023 8th International Conference on Intelligent Information Technology*, pp. 152-156. 2023.
8. I. Indiketiya, K. Kulasekara, J.M. Thomas, I. Gamage and T. Thilakarathna, "DFOG-Image Processing Application for Real-Time Defogging," In *Proceedings of the 2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, Sri Lanka*, 2020, pp. 1-2.
9. H. Farhan and J, Jechang. (2015). Visibility Enhancement of Scene Images Degraded by Foggy Weather Conditions with Deep Neural Networks. *Journal of Sensors* 2016, pp. 1-9.
10. K. O'shea, R. Nash. 2015. "An Introduction to Convolutional Neural Networks." *arXiv* arXiv:1511.08458. November 26, 2015.
11. D. Bank, N. Koenigstein and R. Giryes. 2020. "Autoencoders." *arXiv*.Org. March 12, 2020.
12. O. Ronneberger, P. Fischer and T. Brox. 2015. "U-Net: Convolutional Networks for Biomedical Image Segmentation." *arXiv*.Org. May 18, 2015.
13. K. Jaswinder and K. Prabhpreet. (2017). Comparative study on various single image defogging techniques. In *Proceedings of the 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, pp. 357-361.
14. L. Alzubaidi, J. Zhang and A.J. Humaidi, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J. Big Data 8*, 53 (2021).
15. F. Khozeimeh, D. Sharifrazi and N.H, Izadi, et al. Combining a convolutional neural network with autoencoders to predict the survival chance of COVID-19 patients. *Sci. Rep. 11*, 15343 (2021).
16. Z. Muhammad Hisyam, N. Yudistira and R. Cahya Wihandika. 2022. "Image Colorization Using U-Net With Skip Connections and Fusion Layer on Landscape Images." *arXiv* arXiv:2205.12867, May 25, 2022.
17. R. Dongyu, W. Xiaojun, L. Hui, J. Kittler and X. Tianyang 2021. "UMFA: A Photorealistic Style Transfer Method Based on U-Net and Multi-layer Feature Aggregation." *Journal of Electronic Imaging* 30 (05).
18. T. Zahid, K. Khurshid, A. Salman and K. Khurshid. "Optimisation of transmission map for improved image defogging." *IET Image Processing* 13, no. 7: 1161-1169, 2019.
19. A. Deshmukh, S. Singh and B. Singh, "Design and development of image defogging system," In *Proceedings of the 2016 International Conference on Signal and Information Processing (IConSIP), Nanded, India*, 2016, pp. 1-5.
20. W. Wencheng, Y. Xiaohui, W. Xiaojin, L. Yunlong and S. Ghanbarzadeh. "An efficient method for image dehazing." In *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP)*, 2016; pp. 2241-2245. IEEE..
21. M. Wenjing, Z. Dezhi, C. Minze and C. Juqiang. "Single image defogging via multi-exposure image fusion and detail enhancement." *Journal of Safety Science and Resilience* 5, no. 1: 37-46, 2024.
22. K. Apurva and S. Kumar Sahoo. "A new fast and efficient dehazing and defogging algorithm for single remote sensing images." *Signal Processing* 215: 109289,2024.
23. L. Wei, Z. Fei, L. Tao, D. Jiang and Q. Guoping. "Image defogging quality assessment: Real-world database and method." *IEEE Transactions on Image Processing* 30: 176-190,2020.
24. P. Tannistha, M. Kanti Bhowmik and A, Kumar Ghosh. "Defogging of visual images using SAMEER-TU database." *Procedia Computer Science* 46: 1676-1683,2015.
25. Q. Shuo, W. Zhuowei, Z. Genping, W. Jiahui and Z. Fuhua. "Optical Remote Sensing Image Defogging Algorithm based on Different Wavelength." In *Proceedings of the 2023 15th International Conference on Machine Learning and Computing*, 2023, pp. 406-411.
26. A. Sabiha, M. Ibrahim Khan, M. Md Saki Kowsar, K. Deb, P. Kumar Dhar and T. Koshiba. "Image defogging framework using segmentation and the dark channel prior." *Entropy* 23, no. 3: 285,2021.
27. H. Najmul, S. Ullah, N. Bhatti, H. Mahmood and M. Zia. "A cascaded approach for image defogging based on physical and enhancement models." *Signal, image and video processing* 14, no. 5: 867-875,2020.
28. H. Shengmin, C. Zhixiang, W. Fengli and W. Meiya. "Integrated image defogging network based on improved atmospheric scattering model and attention feature fusion." *Earth Science Informatics* 14, no. 4: 2037-2048, 2021.
29. "Semantic Foggy Scene Understanding With Synthetic Data." n.d. Available at: https://people.ee.ethz.ch/~csakarid/ SFSU_synthetic/ (Accessed on date: 23 March 2024).
30. "Kaggle: Your Machine Learning and Data Science Community." n.d. Available at: https://www.kaggle.com/ (Accessed on date: 26 March 2024).
31. R. Usha & Y. Vamsidhar. (2020). Binary cross entropy with deep learning technique for Image classification. *International Journal of Advanced Trends in Computer Science and Engineering* 9.
32. A. Horé and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM," In *Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey*, 2010, pp. 2366-2369.
33. G. Ahmed Fawzy. 2021. "Accuracy, Precision, and Recall in Deep Learning | Paperspace Blog." Paperspace Blog. April 9, 2021. Available at: https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/ (Accessed on date: 21 March 2024).
34. N. Jim, and T. Akenine-Möller. 2020. "Understanding SSIM." *arXiv* arXiv:2006.13846. June 24, 2020.
35. F.A. Fardo, V.H. Conforto, F.C. de Oliveira and P.S. Rodrigues. "A Formal Evaluation of PSNR as Quality Measurement Parameter for Image Segmentation Algorithms." *arXiv* arXiv:1605.07116. May 23, 2016.