Enhancing Low-Light Images using Deep Learning

MSc Research Project
Data Analytics

Linda Susan Raju
Student ID: x22134409

School of Computing
National College of Ireland

Supervisor: Vladimir Milosavljevic

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Linda Susan Raju
Student ID:	22134409
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Vladimir Milosavljevic
Submission Due Date:	14/12/2023
Project Title:	Enhancing Low-Light Images using Deep Learning
Word Count:	5342
Page Count:	18

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Linda Susan Raju
Date:	31st January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Low-Light Images using Deep Learning

Linda Susan Raju
22134409

Abstract

This research addresses the challenge of low-light image denoising through a multi-stage approach involving convolutional neural networks (CNNs) and a generative adversarial network (GAN) for enhancement. Motivated by the persistent issue of noise in low-light conditions impacting image quality, the study aims to integrate the denoising capabilities of CNN with the refinement offered by a GAN. The CNN models are trained separately to denoise the low-light images. Subsequently, the GAN model is used where its generator component is replaced with the pre-trained denoising CNN model and after the training process, the enhancement using the GAN shows improvements in image quality. It highlights the benefit of the integrated approach compared to the standalone denoising models of sequential processing in achieving low-light image enhancement.

***Keywords*— Low-light images, Denoising, Enhancement, Convolutional Neural Networks (CNN), Generative Adversarial Networks (GANs), Feature Map Based Convolutional Neural Networks (FMBCNN)**

1 Introduction

In the field of imaging technology, the challenge of capturing clear and vibrant images under low-light conditions has become a major area of research. This area holds immense significance across diverse fields such as surveillance, autonomous driving, medical imaging, and photography. The motivation to address this challenge is focused on its broad applications, where the quality and reliability of visual data notably influence decision-making processes. Clean, noise-free images are crucial in these applications, shaping the precision of results and impacting safety, diagnostics, and the overall user experience.

With applications including surveillance, autonomous driving, medical imaging, and photography, research for clear and noise-free images becomes important. The clarity of low-light images directly influences the quality and reliability of results in these domains. Notably, this clarity is essential for object detection and recognition, where precision in visual data is critical for accurate decision-making. The presence of noise and artifacts in low-light conditions can lead to misinterpretations, posing potential risks in contexts like autonomous driving and medical diagnosis. As technology advances, addressing this challenge becomes increasingly crucial for ensuring the integrity of decision-making processes in these high-stakes scenarios. Researchers try to increase the accuracy of findings in an attempt to obtain clear, noise-free images so that decision-makers in a number of fields may depend on precise and understandable visual information.

The resulting improvement in image quality not only caters to preferences in photography but also plays a pivotal role in enhancing safety, security, medical diagnostics, and scientific exploration through clearer and more detailed visual data. The improvement of low-light images in these conditions is evidence of the constant effort to push the limits of innovation for the benefit of society.

1.1 Research Question

How the integration of CNN-based denoising models and GAN-based techniques enhance the quality of low-light images?

1.2 Novelty

Convolutional neural networks (CNN) and generative adversarial networks (GAN) are combined in this research to offer a promising approach for the problems of noise reduction and low-light image enhancement. The proposed method suggests that the denoised CNN model's ability to capture intricate features is refined through GAN training, resulting in enhanced visual quality in low-light conditions; that is, the denoised CNN model will be used as a generator in the GAN model to produce even more enhanced output. From all the denoised models, the Feature Map Based Convolutional Neural Network (FMBCNN), which uses the inherent ability of feature maps to capture detailed visual features and patterns, further distinguishes this research. Given the importance of low-light images in applications, the primary focus on improving low-light image quality using denoising processes is recent. The empirical data showing that the FMBCNN approach outperforms traditional filtering methods is the new factor in this research, showing its capacity to significantly improve the area of low-light image denoising.

2 Related Work

2.1 Denoising CNN Approach

Ilesanmi & Ilesanmi (2021) provides an overview of the techniques in image denoising, with a focus on convolutional neural network (CNN) methods. It addresses the importance of image denoising due to the increase of digital images captured in poor conditions, emphasizing its importance in various domains such as medical imaging, remote sensing, and forensics. The paper categorizes traditional filters, including linear, non-linear, adaptive, wavelet-based, partial differential equation (PDE), and total variation filters, and highlights their limitations, such as poor test phase optimization and manual parameter settings. This is where they introduce CNN as a flexible solution to these drawbacks. It also discusses the evaluation metrics for the analysis of CNN image denoising performance, which are mean square error (MSE), peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), root mean square error (RMSE), feature similarity (FSIM and FSIMc), and signal-to-noise ratio (SNR).

The study identifies other CNN techniques, such as the feedforward CNN, U-Net, and residual networks, while also highlighting challenges, including limited memory for CNN applications and difficulties in solving unsupervised denoising tasks. The paper provides an understanding of CNN-based image denoising methods and promotes further exploration in this field.

In CNN, feature maps are essential components that are generated by applying a convolution operation to an input image using specific filters. These maps represent various aspects of the input image, with early layers extracting simple features like edges and later layers identifying more complex features such as corners and textures. Feature maps play a crucial role in helping the network understand the relationships between features in an image, which leads to more accurate projections and improved performance. Ravi et al. (2023) presents FMBCNN, a feature map-based convolutional neural network for image denoising. FMBCNN increases channel interdependencies in CNNs without significant computation costs and dynamically selects features using feature maps. It outperforms other denoising techniques in terms of metrics by its success in achieving a lower Mean Square Error (MSE) of 70.105, a higher Peak Signal-to-Noise Ratio (PSNR) of 31.696, a superior Structural Similarity Index (SSIM) of 0.924, and an entropy of 0.443. It also manages both artificial and real noise. The study emphasizes the importance of feature maps in CNN-based denoising and suggests that FMBCNN is a promising method for image denoising.

Islam et al. (2018) addresses the challenging problem of mixed-noise removal in images, focusing on the common combination of additive white Gaussian noise (AWGN) and impulse noise (IN). Despite the non-linearity in noise distribution, the paper introduces an algorithm using a convolutional neural network (CNN) model for effective denoising. The CNN model, adopting a computationally efficient transfer learning approach, establishes an end-to-end mapping from noisy to noise-free images. Despite its compact structure, the CNN model outperforms established methods in accuracy and robustness. Experimental results across diverse mixed-noise settings confirm the effectiveness of the CNN-based denoising method, highlighting both efficiency and superior performance. The proposed method allows for faster denoising operations compared to previous methods. Overall, it presents a promising solution for effectively reducing mixed Gaussian-impulse noise from images, offering improved accuracy, robustness, and computational efficiency compared to existing methods.

2.2 GAN Enhancement Approach

Xu et al. (2022) focuses on an underwater image enhancement algorithm employing an improved GAN model, which addresses challenges in obtaining paired training sets for supervised learning. The proposed method integrates the GAN model with a global-local discriminator structure, introducing a Wasserstein-GAN with gradient penalty loss and combining L1 and L2 loss functions for enhanced performance. The algorithm utilizes a U-Net structure with an Adaptive Dense Feature Fusion (ADFF) module to effectively retain and accumulate key features from various levels, aiming to improve cross-scale connections and multi-level feature integration. Experimental analysis on a dataset of 3,800 underwater and land images demonstrates the algorithm's superior performance compared to classic methods, as evaluated by objective indicators such as UIQM, UIConM, UCIQE, and information entropy. The study concludes by highlighting the algorithm's capacity to restore underwater images with fine details and natural colours, suggesting potential for further work in addressing details of image enhancement, particularly in overexposed areas.

Medical image enhancement is a critical aspect of pre-processing in automated analysis and diagnosis. The challenges posed by different conditions across different imaging devices necessitate advanced techniques to produce high-quality images for accurate clinical interpretation. Ma et al. (2021) reveals a gap in achieving consistent and detailed enhancement across various medical imaging modalities, and StillGAN addresses this gap through an unpaired learning framework, treating low- and high-quality images as distinct domains. By incorporating novel constraints on illumination and structure loss, StillGAN demonstrates superiority over existing methods, particularly in capturing local details crucial for clinical interpretation. The experiments on corneal confocal microscopy and colour fundus images showcase the method's effectiveness in improving SNR, nerve fibre segmentation, and overall visual perception.

However, limitations are identified, such as challenges in handling non-uniform intensity regions and the risk of incorrect translation, raising the need for further improvement in structure loss. The paper suggests that refining these constraints and incorporating colour consistency considerations may enhance StillGAN's adaptability to different medical imaging conditions. It emphasises the significance of medical image enhancement in facilitating accurate diagnosis and therapy planning. StillGAN, with its unique constraints, stands out as a promising solution to address the limitations of existing methods.

2.3 Integrated Approaches

Vashisht et al. (2023) and Shi et al. (2022) represent significant contributions to the realms of medical image categorization, hemolysis image identification, and PD pattern recognition within GIS. They employ a fusion of CNN and GAN methodologies, each with distinct applications and approaches. Vashisht et al. (2023), the focus lies on medical image classification, specifically pneumonia, utilising CNN-GAN methods. This study underscores the potency of GAN-driven data augmentation for dataset enhancement, effectively tackling challenges in medical image classification. Conversely, Shi et al. (2022) concentrates on hemolysis detection in medical images, employing CNN-GAN techniques for both data expansion and feature extraction. This highlights the versatility of CNN-GAN beyond classification tasks. Wang et al. (2022), the emphasis is on PD pattern recognition in GIS with imbalanced samples. This research utilizes GAN and CNN to rectify data imbalances and automatically optimize CNN construction, showcasing the adaptability of CNN-GAN in real-world data scenarios and model optimization for GIS pattern recognition tasks.

Vashisht et al. (2023) underscores the importance of GAN-based data augmentation in medical image classification, specifically pneumonia. This aligns with the potential advantages of GAN-based data augmentation in the current study, offering benefits in creating diverse training data to manage variations in low-light conditions and noise levels effectively.

3 Methodology

In this section, a detailed overview of the methodology used in this project with the key approaches and techniques is highlighted. The methodology outlines the gradual process followed to develop and evaluate CNN and GAN models for the denoising and enhancement tasks.

3.1 Dataset Collection

The LOL (LOw-Light) dataset is a repository comprising 500 low-light and normal-light image pairs, which have separate training and testing pairs along with same distribution making this a balanced dataset. Each image in the dataset has a resolution of 400x600 and was captured in indoor areas. Figure 1 and Figure 2 show the low-light and its corresponding normal-light images from the LOL dataset.

Figure 1: Low-light images in LOL dataset

Figure 2: Corresponding Normal-light images in LOL dataset

3.2 Data Splitting and Preparation

During the preprocessing step, the function 'load_and_preprocess_images', is designed to handle the loading and preprocessing of images. This function operates on a specified folder path containing low-light and normal-light images for subsequent processing. First, the image files are sorted since the pairs may not share the same index. After sorting, the images are resized and then normalised to a pixel value range of [0, 1] to ensure consistency in model input. Here, data augmentation is not done; only resizing and normalisation are applied to the images since data augmentation increases the computational load, and limited computation is one of the drawbacks of this work. So, the number of images used for training were limited.

Since the dataset has separate training and testing folders with 485 and 15 images, respectively, in each folder for both low-light and normal-light images. The training images are split using the 'train_test_split' function from Scikit-learn, to training and validation with 388 and 97 images respectively.

3.3 CNN Model Architectures

Figure 3 shows the outline of all the CNN models used in this research.

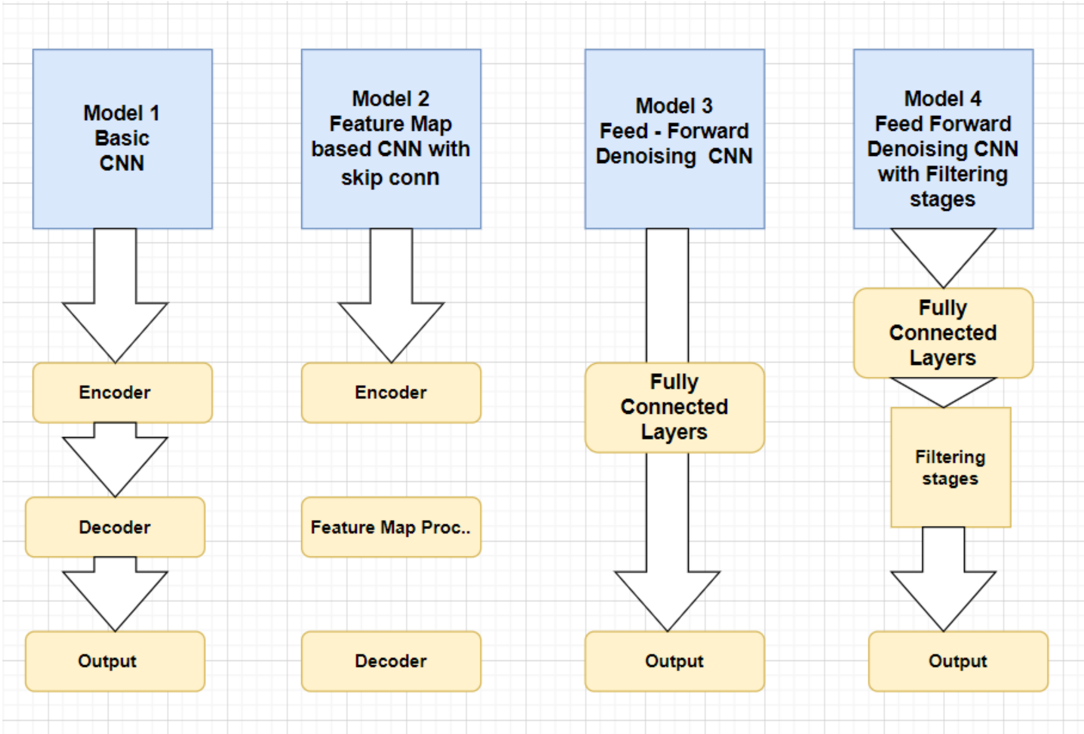

Figure 3: Four CNN Models

3.3.1 Model 1: Basic CNN Model

The architecture of the first model designed is a basic Convolutional Neural Network (CNN) for the purpose of denoising low-light images. The model is constructed using the Keras Sequential API, encompassing an encoder-decoder structure. The use of convolutional and transpose convolutional layers enables the model to capture and reconstruct essential features for effective low-light image denoising.

The encoder begins with a convolutional layer comprising 64 filters with a (3, 3) kernel size, employing the rectified linear unit (ReLU) activation function and same-padding. Batch normalisation is applied to enhance training stability. Subsequently, a similar pattern is repeated with a convolutional layer featuring 128 filters. A MaxPooling layer with a (2, 2) pool size is then employed to down-sample the spatial dimensions.

The decoder mirrors the encoder architecture, utilising Conv2DTranspose layers for up-sampling. This begins with a Conv2DTranspose layer with 128 filters, followed by another with 64 filters. Batch normalisation is applied to each layer for stability. Finally, an UpSampling2D layer with a (2, 2) size is utilised to restore the spatial dimensions.

The output layer consists of a Conv2D layer with three filters, employing the sigmoid activation function and same-padding. This architecture is tailored for denoising tasks, aiming to reconstruct denoised low-light images.

3.3.2 Model 2: Feature-map Based CNN with Skip Connections

The second model uses a feature-map-based Convolutional Neural Network (CNN) architecture enhanced by the incorporation of skip connections, similar to the concept of Ravi et al. (2023). This model is specifically designed for denoising low-light images, using both downsampling and upsampling pathways. The model begins with an input layer, accepting low-light images with a specified shape of (400, 600, 3). The encoding pathway commences with a series of convolutional layers. The first convolutional layer employs 64 filters with a kernel size of 3 and utilises batch normalisation and rectified linear unit (ReLU) activation. This is followed by a similar pattern with 128 filters and subsequent down-sampling using strides of 2. The process continues with a layer featuring 256 filters and further down-sampling.

The feature map processing layer involves upsampling and concatenation of skip connections. The feature maps from the encoding pathway are upsampled, and skip connections from previous layers are concatenated to preserve high-level features. A convolutional layer with 128 filters is then applied.

The decoding pathway mirrors the encoding process, but in reverse. Upsampling is performed, and skip connections are concatenated to the feature maps at each step. This enhances the model's ability to capture fine-grained details during image reconstruction.

The final output layer consists of a convolutional layer with three filters, employing the sigmoid activation function. The utilisation of skip connections enables the model to effectively capture and retain important features, enhancing the denoising capability.

3.3.3 Model 3: Feed-Forward Denoising CNN

This model is a feed-forward denoising Convolutional Neural Network (CNN) architecture. A feed-forward denoising Convolutional Neural Network (CNN) is a type of neural network designed to remove noise from input data, especially images. Unlike the previous models that incorporated convolutional layers for spatial feature extraction, this model relies on fully connected layers to process and denoise low-light images. The model begins with an input layer designed to accommodate low-light images of a specified shape. The input is flattened, converting the multi-dimensional image data into a one-dimensional vector. This step facilitates the processing of the image content through fully connected layers.

The flattened input is connected to dense layers for feature processing. The model incorporates two dense layers with 1024 and 512 neurons, respectively, each activated by the rectified linear unit (ReLU) activation function. These layers play a crucial role in learning hierarchical representations of the input data.

The output layer consists of a dense layer with the number of neurons equal to the product of the input shape. The sigmoid activation function is employed to ensure output values fall within the [0, 1] range. This layer aims to reconstruct denoised low-light images from the learned representations.

To restore the output to the original image shape, a reshape layer is introduced, which is crucial for preserving the spatial structure of the denoised images. The feed-forward denoising CNN model represents an alternative approach to low-light image denoising, relying on fully connected layers for feature processing. This architecture provides flexibility in handling diverse image characteristics and contributes to the overall diversity of our denoising model ensemble.

3.3.4 Model 4: Feed-Forward Denoising CNN with Filtering Stages

This model also uses a feed-forward denoising Convolutional Neural Network (CNN) architecture, which is distinguished by a unique multi-stage filtering approach, a concept from Islam et al. (2018). The initial layer of the model encompasses a convolutional layer with 64 filters, each of size 3x3, utilizing batch normalization and rectified linear unit (ReLU) activation. The core of the model consists of four consecutive convolutional filtering stages. Each stage contains a sequence of convolutional layers, batch normalisation, ReLU activation, max pooling, and upsampling operations. This intricate combination of operations aims to extract hierarchical features at different spatial resolutions, promoting effective denoising. Following the filtering stages, a global average pooling layer is employed to capture the overall spatial information, reducing the spatial dimensions and summarizing the learned features. A fully connected layer with 256 neurons and ReLU activation serves as the bridge between the convolutional layers and the final output layer. This layer facilitates the aggregation and processing of high-level feature representations.

The output layer comprises a dense layer with the number of neurons equal to the product of the input shape. The sigmoid activation function is applied to ensure output values are confined within the [0, 1] range. The final reshape layer reconstructs the denoised low-light images to their original shape. The unique aspect of this model lies in its multi-stage filtering strategy, offering a distinctive approach to low-light image denoising.

3.3.5 Hyperparameter Tuning

To optimise the performance of the denoising models, hyperparameter tuning is a critical step in the methodology. The objective is to evaluate various combinations of hyperparameters and identify the configuration that yields optimal results in terms of Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM). The 'evaluate_cnn_model' function is designed for each model to assess the denoising model's performance based on a set of hyperparameters. The function builds the CNN denoising model using the specified hyperparameters, compiles it with the Adam optimizer, and trains it on the training set. Subsequently, the model's performance is evaluated on the validation set using MSE, PSNR, and SSIM metrics.

A hyperparameter grid, denoted by 'param_grid', is defined, encompassing different values for learning rates, epochs, and batch sizes. This grid serves as the search space for hyperparameter combinations. All possible combinations of hyperparameters are generated using the 'ParameterGrid' function, resulting in a list of dictionaries (‘param_combinations‘), each representing a unique set of hyperparameters.

The ‘evaluate_cnn_model’ function is executed for each hyperparameter combination within the ‘param_combinations‘ list. The results, including the specific hyperparameters, MSE, PSNR, and SSIM, are then printed for analysis. By considering a range of learning rates, epochs, and batch sizes, we can strike a balance between model performance and computational efficiency.

3.3.6 Training Procedure

In the training phase of the denoising model, the procedure is outlined as follows:

- A denoising model is initialised using the architecture defined in the build CNN model function.
- The model is compiled with all the hyperparameters obtained from the hyperparameter tuning phase. The Adam optimizer is utilised with the specified learning rate, and Mean Squared Error (MSE) is employed as the loss function. Additionally, accuracy is tracked as a metric to assess model performance along with MSE, PSNR, and SSIM.
- Two essential callback functions are employed during training. The 'ModelCheckpoint' callback is set to save the best model weights during training, ensuring that the model with the lowest validation loss is retained. The 'EarlyStopping' callback monitors the validation loss and terminates training if no improvement is observed after a certain number of epochs (patience). This prevents overfitting and accelerates convergence.
- The training is executed over a specified number of epochs, with the batch size during hyperparameter tuning. The validation data is used to monitor the model's performance on unseen data during training.
- The training history, including metrics such as loss and accuracy, is captured in the 'history' object. This information provides insights into the model's convergence and performance trends.
- The denoised images of each model for different hyperparameters are then generated and displayed.
- The trained model's weights are saved using the 'ModelCheckpoint' callback to ensure that the best-performing model is retained for subsequent use.

This structured training procedure ensures that the denoising model is effectively trained, leveraging optimal hyperparameters and incorporating mechanisms for model checkpointing and early stopping to enhance performance and prevent overfitting.

3.4 GAN Model Architecture

The GAN model used is a conditional Generative Adversarial Network (GAN) architecture for image enhancement, specifically targeting low-light image denoising. It consists of a generator and a discriminator, each with specific architectural configurations. The training process involves adversarial training, where the generator aims to produce high-quality images that are indistinguishable from real high-light images, and the discriminator learns to differentiate between real and generated images.

3.4.1 Generator Architecture

The generator is responsible for transforming low-light images into high-quality images. It takes low-light images as input with the specified shape (400, 600, 3). The encoder uses 64 filters of convolutional layers along the rectified linear unit (ReLU) activation function with batch normalization and again 128 filters of convolutional layers with batch normalization. There are six residual blocks with batch normalisation, and each block uses the rectified linear unit (ReLU) activation function.

The decoder employs 64 filters of convolutional layers with batch normalisation and concludes with a convolutional layer using 'sigmoid' activation for image output. The final output is obtained by adding the generated image to the input, enhancing low-light images.

3.4.2 Discriminator Architecture

The discriminator evaluates the authenticity of images, distinguishing between real and generated ones. This model takes either real or generated images as input and utilises 64 and 128 filters of convolutional layers with the rectified linear unit (ReLU) activation function. It then flattens the output and connects to a dense layer with 'sigmoid' activation for binary classification.

3.4.3 Residual Block

This block takes feature maps as input and uses convolutional layers with batch normalisation with the activation function. It has skip connections to add the input feature maps to the output of the convolutional layers and then apply the activation function to the combined output.

3.4.4 Training Procedure

This is the part where combining the CNN and GAN happens; the GAN model uses the denoised CNN model as the generator and then builds the discriminator. To maintain the integrity of the pre-trained denoised model, its weights are frozen during GAN training. Freezing prevents the generator (denoised model) from updating its weights based on the adversarial training process, ensuring it retains the knowledge acquired during the denoising training. The GAN model is compiled with the denoised model as the generator and the newly constructed discriminator. The aim is to produce images identical to normal light images and to fool the discriminator. Binary cross-entropy loss is used as the objective function, as GANs involve a binary classification task for the discriminator (real or generated).

3.4.5 Hyperparameters

The hyperparameter tuning in GAN is the same as that of CNN hyperparameter tuning; a hyperparameter grid is defined with different learning rates, epochs, and batch sizes. The evaluation results, including metrics such as Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM), are then printed.

3.4.6 Generated Images

A function (‘save_generated_images‘) is defined to save and display low-light images, corresponding high-light images, and the generated images, allowing a qualitative assessment of the generator's output.

4 Implementation

The proposed solution for low-light image denoising and enhancement was executed, which involved the independent development and training of CNN models. During the training of the GAN model, the denoised CNN model acted as its generator component, with the aim of generating identical mormal light images once the GAN was compiled. For finding the better model, the focus was on achieving optimal model performance through hyperparameter tuning and other evaluation metrics.

4.1 Evaluation Metrics

Quantitative assessment of the models' performance was carried out using evaluation metrics such as Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM). These metrics provided numerical insights into the effectiveness of the models in denoising low-light images. Python, supplemented by libraries like scikit-image, was employed for the calculation of these metrics.

4.2 CNN Models

The CNN models were designed to address the low-light image denoising task. All the CNN models underwent hyperparameter tuning for batch size, number of epochs, and learning rate. TensorFlow and Keras in Python served as the principal tools for model development and training. The outcome of this phase was a set of trained models, each associated with specific hyperparameter configurations.

Tables below show the results of all the models with their various combinations of batch size (under column Batch), epochs, and learning rates (LR).

4.2.1 Model 1: Basic CNN Model

The hyperparameters that yield the best results in this model include a learning rate of 0.01, 50 epochs, and a batch size of 16. Upon evaluating the model's performance on the test dataset, the test loss is 0.0204 and the accuracy is 67.21%. This specific set of hyperparameters contributes to the model's capacity to effectively reduce noise and enhance image quality within the denoising context. Figure 4 shows the generated images of this CNN model.

Table 1: Basic CNN Model Performance with Different Hyperparameters

#	Batch	Epochs	LR	MSE	PSNR	SSIM	Metrics
1	8	50	0.0001	0.184	6.765	0.263	loss: 0.0200 - accuracy: 0.6646
2	8	50	0.001	0.173	6.281	0.254	loss: 0.0202 - accuracy: 0.6133
3	8	50	0.01	0.208	4.970	0.226	loss: 0.0210 - accuracy: 0.7671
4	16	50	0.0001	0.169	6.162	0.253	loss: 0.0195 - accuracy: 0.6672
5	16	50	0.001	0.173	5.961	0.260	loss: 0.0197 - accuracy: 0.6543
6	16	50	0.01	0.157	7.357	0.284	loss: 0.0204 - accuracy: 0.6721

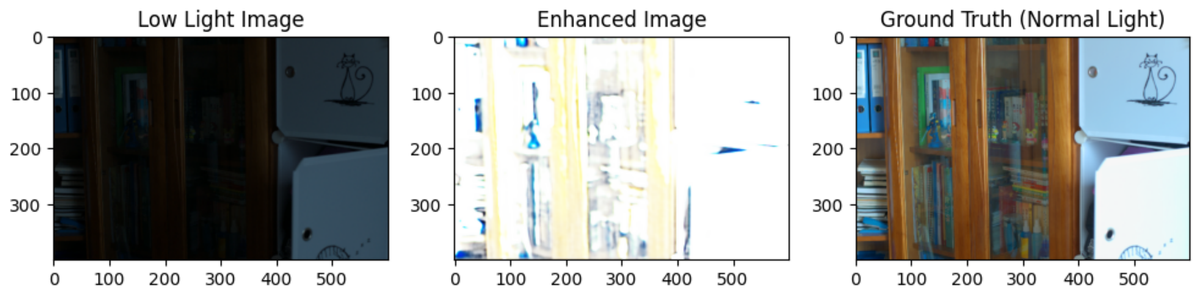

Figure 4: Basic CNN Model's Generated Images

4.2.2 Model 2: Feature-map Based CNN with Skip Connections

The optimal hyperparameters identified in this model include a learning rate of 0.0001, 50 epochs, and a batch size of 8. Upon evaluating the model's performance on the test dataset, the test loss is 0.0205, the accuracy is 71.67%, and it has the highest PSNR value of 7.237 compared to other combinations. This specific set of hyperparameters contributes to the model's ability to capture intricate features through its feature-map-based architecture with skip connections, resulting in exceptional denoising outcomes. Figure 5 shows the generated images of this FMBCNN model.

Table 2: FMBCNN Model Performance with Different Hyperparameters

#	Batch	Epochs	LR	MSE	PSNR	SSIM	Metrics
1	8	50	0.01	0.171	6.668	0.282	loss: 0.0249 - accuracy: 0.7065
2	8	50	0.001	0.176	6.805	0.278	loss: 0.0230 - accuracy: 0.7404
3	8	50	0.0001	0.183	7.237	0.281	loss: 0.0208 - accuracy: 0.7167
4	8	100	0.01	0.199	6.088	0.268	loss: 0.0265 - accuracy: 0.7112
5	8	100	0.001	0.181	6.404	0.272	loss: 0.0206 - accuracy: 0.7600
6	8	100	0.0001	0.173	6.857	0.285	loss: 0.0206 - accuracy: 0.7131
7	16	50	0.01	0.167	6.09	0.284	loss: 0.0228 - accuracy: 0.6565
8	16	50	0.001	0.195	5.985	0.263	loss: 0.0210 - accuracy: 0.7183
9	16	50	0.0001	0.167	6.654	0.279	loss: 0.0201 - accuracy: 0.6787
10	16	100	0.01	0.179	7.150	0.297	loss: 0.0275 - accuracy: 0.6989
11	16	100	0.001	0.168	6.510	0.276	loss: 0.0197 - accuracy: 0.7542
12	16	100	0.0001	0.171	6.619	0.272	loss: 0.0188 - accuracy: 0.7004

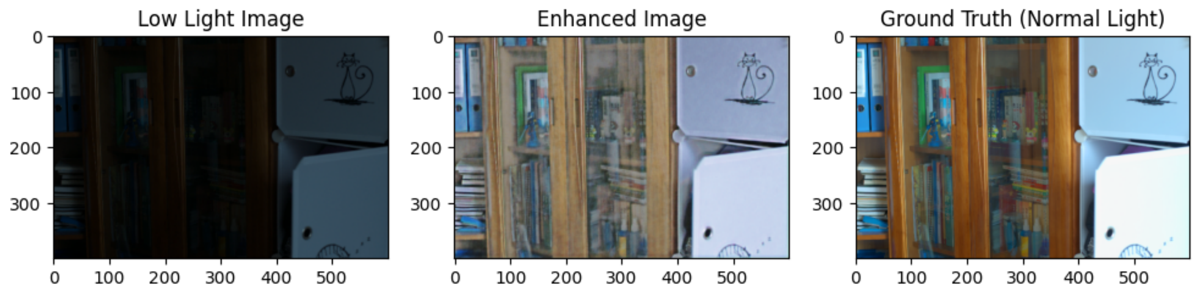

Figure 5: FMBCNN Model's Generated Images

4.2.3 Model 3: Feed-Forward Denoising CNN

The identified best hyperparameters for this model are a learning rate of 0.001, 50 epochs, and a batch size of 16. Upon evaluating the model's performance on the test dataset, the test loss is 2.26% and the accuracy is 76.04%.

The test loss of 0.0226 represents the mean squared error between the denoised images and the ground truth images. Additionally, the accuracy of 76.04% indicates the proportion of correctly denoised pixels, showcasing the model's ability to generalise well to unseen data. Figure 6 shows the generated images of this feed-forward model.

Table 3: Feed-Forward Denoising Performance with Different Hyperparameters

#	Batch	Epochs	LR	MSE	PSNR	SSIM	Metrics
1	8	50	0.0001	0.187	6.878	0.346	loss: 0.0175 - accuracy: 0.6907
2	8	50	0.001	0.190	6.724	0.326	loss: 0.0207 - accuracy: 0.6923
3	8	50	0.01	0.157	4.09	0.330	loss: 0.0574 - accuracy: 0.5640
4	16	50	0.0001	0.180	7.511	0.333	loss: 0.0216 - accuracy: 0.6282
5	16	50	0.001	0.214	7.043	0.324	loss: 0.0226 - accuracy: 0.7604
6	16	50	0.01	0.156	4.147	0.332	loss: 0.0574 - accuracy: 0.5614

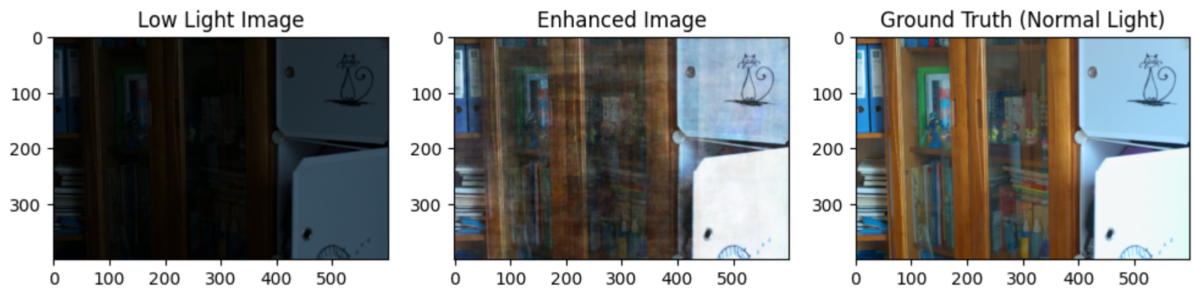

Figure 6: Feed-Forward CNN Model's Generated Images

4.2.4 Model 4: Feed-Forward Denoising CNN with Filtering Stages

Among the tested configurations, this model doesnt́ provide the best results; the generated image is blurry, and the highest set of hyperparameters consists of a learning rate of 0.001, 50 epochs, and a batch size of 16. Upon evaluating the model's performance on the test dataset, the test loss is 0.0638 and the accuracy is 45.36%. Figure 7 shows the generated images of this feed-forward with filters CNN model.

Table 4: Feed-Forward Denoising Performance with Different Hyperparameters

#	Batch	Epochs	LR	MSE	PSNR	SSIM	Metrics
1	8	50	0.0001	0.142	2.597	0.322	loss: 0.0502 - accuracy: 0.5400
2	8	50	0.001	0.150	4.842	0.332	loss: 0.0524 - accuracy: 0.4857
3	8	50	0.01	0.106	10.827	0.441	loss: 0.0820 - accuracy: 0.2511
4	16	50	0.0001	0.0693	4.315	0.497	loss: 0.0612 - accuracy: 0.498
5	16	50	0.001	0.0475	11.7514	0.5837	loss: 0.0638 - accuracy: 0.4536
6	16	50	0.01	0.0496	10.5300	0.5489	loss: 0.0596 - accuracy: 0.4731

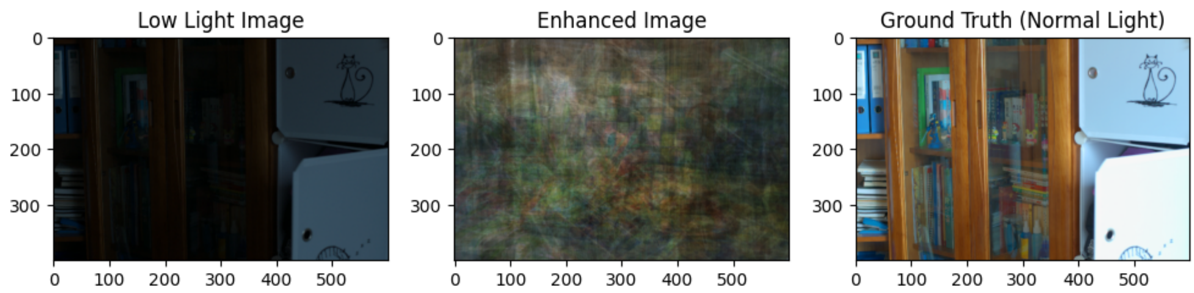

Figure 7: Feed-Forward with Filters CNN Model's Generated Images

4.3 GAN Model

The GAN model architecture consisted of a generator and a discriminator, both featuring specific layers and configurations. The denoised output from the CNN model serves as the input for the GAN model. The GAN model generates enhanced images based on the denoised input, aiming to further improve image quality. The GAN model leverages the pre-trained denoised model as its generator, emphasising the sequential processing of denoising and enhancement. The discriminator distinguishes between real and generated images, contributing to the adversarial training process; it is evaluated by its loss and accuracy. The outputs are analysed, comparing the generated enhanced images with ground-truth high-light images.

5 Evaluation

This research showcases an exploration of the integration of Convolutional Neural Networks (CNN) and Generative Adversarial Networks (GAN) for addressing the challenges of noise reduction and low-light image enhancement. To analyse the better denoising model, four CNN models were executed, and by using the metrics PSNR and accuracy, it was found that, compared to all the denoising models, the FMBCNN denoised model had the highest PSNR of 7.15 and an accuracy of 69.89%. Though other models had an accuracy higher than that, for denoising, PSNR and SSIM are the important evaluation metrics. The analysis involved a rigorous evaluation of hyperparameter combinations, revealing optimal configurations that significantly improve denoising accuracy.

Figure 8 shows the resulting denoised image of integrated basic CNN model and GAN, which has a PSNR value of 15.93 and an SSIM of 0.811, alongside their corresponding original images for qualitative assessment.

Figure 8: Output of the integrated basic CNN and GAN model

Figure 9 shows the resulting denoised image of integrated feed-forward CNN model and GAN, which has a PSNR value of 17.18 and an SSIM of 0.760, alongside their corresponding original images for qualitative assessment.

Figure 9: Output of the integrated feed-forward CNN and GAN model

Figure 10 shows the resulting denoised image of integrated feed-forward CNN model with filtering stages and GAN, which has a PSNR value of 12.21 and an SSIM of 0.715, alongside their corresponding original images for qualitative assessment.

Figure 10: Output of the integrated feed-forward CNN with filtering stages and GAN model

This study not only demonstrates the performance of the FMBCNN approach over traditional filtering methods but also emphasizes the collaborative benefits of sequential processing with CNN for denoising and GAN for image enhancement. Figure 11 shows the resulting denoised image of integrated FMBCNN and GAN model, which has a PSNR value of 17.43 and an SSIM of 0.888, visualised alongside their corresponding original images for qualitative assessment.

Figure 11: Output of the integrated FMBCNN and GAN model

The enhanced images of the chosen model exhibit improved clarity and preservation of essential details compared to their original low-light counterparts. The incorporation of skip connections in the architecture contributes to the model's ability to capture and retain intricate features during the denoising process.

The feature-map-based CNN architecture, enriched by skip connections, proves to be a robust solution for addressing the challenges posed by low-light conditions. These findings highlight the significance of architectural choices in image denoising tasks and the potential of this model in real-world applications where low-light image quality is a critical factor.

The FMBCNN model served as the generator component of the GAN model, keeping the denoising model's pre-trained data and then generating enhanced images for a cascaded approach to image enhancement. This sequential processing aimed to refine the denoised images further, capturing subtle details and improving perceptual quality. The GAN's ability to learn and adapt to the specific characteristics of the denoised images was crucial in achieving the desired enhancements.

The combination of the FMBCNN model and the GAN introduced a multi-stage approach to low-light image processing. The FMBCNN model, with its feature-map-based architecture, effectively reduced noise and preserved important features. The subsequent enhancement by the GAN demonstrated its role in refining denoised images, addressing potential artefacts, and enhancing visual appeal.

5.1 Limitations

Despite getting a refined, denoised image, there is still scope for improvement in the performance of the model. There are a few limitations to this project, one of which is computation. All four denoising models are executed in separate files to avoid any GPU memory issues, but still, during hyperparameter tuning, when the batch size increases, the GPU runs out of memory, making it different to work on larger batch sizes and epochs and limiting the models' performance. Similarly, for this reason, a small dataset and less complex denoising models were used.

6 Conclusion and Future Work

In conclusion, this research sets out to address the challenges of noise reduction and low-light image enhancement through the integration of Convolutional Neural Networks (CNN) and Generative Adversarial Networks (GAN). The objective was to improve the quality of low-light images using denoising and enhancement processes. The results have demonstrated the FMBCNN approach performed better over traditional filtering methods, showcasing its capability to significantly enhance low-light image denoising. The research emphasises the collaborative benefits of sequential processing, where the denoised CNN model serves as the generator for the GAN model, resulting in enhanced image quality. This approach proves effective in capturing intricate features and patterns, surpassing traditional methods. However, the study acknowledges several limitations, including potential data bias and resource-intensive computations. These limitations prompt a critical reflection on the implications of the research findings. Practical applications include improved image analysis accuracy in surveillance systems, highlighting the real-world impact of AI advancements.

Future work can involve evaluations of the GAN model with alternative architectures and hyperparameter configurations. The potential for commercialization is evident, suggesting opportunities for user-friendly tools or software aimed at a broader audience. This paves the way for innovative applications in various industries, marking a step towards the practical implementation of research findings.

References

Ilesanmi, A. E. & Ilesanmi, T. O. (2021), 'Methods for image denoising using convolutional neural network: a review', *Complex & Intelligent Systems* **7**(5), 2179–2198.

Islam, M. T., Rahman, S. M., Ahmad, M. O. & Swamy, M. (2018), 'Mixed gaussian-impulse noise reduction from images using convolutional neural network', *Signal Processing: Image Communication* **68**, 26–41.

Ma, Y., Liu, J., Liu, Y., Fu, H., Hu, Y., Cheng, J., Qi, H., Wu, Y., Zhang, J. & Zhao, Y. (2021), 'Structure and illumination constrained gan for medical image enhancement', *IEEE Transactions on Medical Imaging* **40**(12), 3955–3967.

Ravi, A., Dupaguntla, G., Tadepalli, D. K., Vissamsetti, S. & Achanta, R. K. (2023), Image denoising using feature map based convolutional neural networks, *in* '2023 International Conference on Recent Trends in Electronics and Communication (ICRTEC)', pp. 1–5.

Shi, X., Deng, Y., Fang, Y., Chen, Y., Zeng, N., Fu, L. et al. (2022), 'A hemolysis image detection method based on gan-cnn-elm', *Computational and Mathematical Methods in Medicine* **2022**.

Vashisht, S., Lamba, S., Sharma, B. & Sharma, A. (2023), Pneumonia classification using cnn-gan, *in* '2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)', IEEE, pp. 456–461.

Wang, Y., Yan, J., Yang, Z., Jing, Q., Wang, J. & Geng, Y. (2022), 'Gan and cnn for imbalanced partial discharge pattern recognition in gis', *High Voltage* **7**(3), 452–460.

Xu, B., Zhou, D. & Li, W. (2022), 'Image enhancement algorithm based on gan neural network', *IEEE Access* **10**, 36766–36777.