# Denoising Fourier Noise using REDNet in Image

Thinh Duong Tan Hung*
FPT University
Thu Duc, Viet Nam
thinhdthse172922@fpt.edu.vn

Loc Phan Minh
FPT University
Thu Duc, Viet Nam
locpmse161801@fpt.edu.vn

## ABSTRACT

Fourier noise, often referred to as frequency domain noise, is a particular kind of interference or noise that affects data, pictures, audio, and digital signals in the frequency domain. The Fourier transform—a basic mathematical tool for the study and manipulation of signals in the frequency domain—was invented by renowned mathematician and scientist Jean-Baptiste Joseph Fourier, who is honored by the name of this noise. However, existing method for denoising Fourier Noise such as FFT has a significant drawback. FFT can not automatically denoise Fourier, we have to adjust it. Therefore, this method still not the best solution for denoising Fourier Noise. To address this challenge, our study proposes a novel approach that is Residual Encoder-Decoder Network(REDNet). It is a deep neural network architecture developed specifically for problems involving picture restoration, particularly denoising. The result have PSNR, MSE are 5,72 and 104,52 respectively. These result indicated REDNet is designed for denoising image and work well on denoising Fourier

## CCS CONCEPTS

• **Computing methodologies → Artificial Intelligence**.

## KEYWORDS

REDNet, Image Converting, Denoising, Deep Learning, Skip Connection, Fourier

**ACM Reference Format:**
Thinh Duong Tan Hung and Loc Phan Minh. 2018. Denoising Fourier Noise using REDNet in Image. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

## 1 INTRODUCTION

The first step in the suggested strategy is to train a specific REDNet model designed for Fourier noise denoising. This model captures and restores the underlying visual features by using an encoder-decoder structure with residual connections. The decoder portion reconstructs a clear and aesthetically acceptable image, while the encoder portion gathers important characteristics from the noisy Fourier-transformed input. The capacity of REDNet to manage intricate picture attributes and long-range relationships, which are essential for efficiently combating Fourier Noise, is one of this solution's main advantages. Suppressing the undesirable frequency domain changes associated with Fourier Noise is an excellent capability of the REDNet model, which learns the residual information between the noisy input and the intended output. An extensive evaluation of the suggested method is conducted using a variety of picture types and Fourier Noise levels. The outcomes show off REDNet's outstanding denoising skills, with notable gains in image quality and a notable decrease in Fourier Noise distortions. This novel approach holds considerable potential for a number of applications, including digital signal processing, medical imaging, and astronomical observations, where it is necessary to restore the quality of pictures impacted by frequency domain noise.

A particular kind of deep neural network architecture called REDNet, or Residual Encoder-Decoder Network, is intended for picture denoising applications. Its many advantages make it a well-liked option for picture denoising. It has been demonstrated that REDNet topologies are quite good at eliminating noise from pictures. They represent the intricate relationships in noisy and clean pictures using a deep neural network that has encoder and decoder components, which improves denoising performance. The foundation of REDNet is residual learning, which is the process of determining the residual, or difference, between noisy and clean pictures. By assisting the network in concentrating on simulating noise, residual learning improves denoising efficiency. REDNet extracts features from the data itself, as opposed to creating them by hand. This increases its versatility by enabling it to adjust to different kinds and volumes of noise. REDNet topologies are frequently made to run in parallel, which makes training and inference faster—especially on contemporary hardware that has many GPUs. REDNet's deep design allows it to effectively capture tiny features and complex noise patterns, which is useful for high-quality picture denoising. Since REDNet models are trained end-to-end, noisy pictures are fed into the model, and denoised images are produced. This streamlines the denoising procedure and may produce improved outcomes. It is possible to modify and adjust REDNet topologies to suit various noise levels and picture kinds. The model's architecture and training set may be modified to better fit certain denoising requirements. Therefore, this model is designed for denoising image.

The remainder of this paper is structured as follows: We review previous studies in the "Related Work" section. In "Method," we provide detailed information about our methodology. The "Experiment" section presents and analyzes the experimental results. Additionally, we offer an in-depth explanation of our approach, discussing its merits, limitations, and potential applications in the "Discussion" section. Finally, we conclude and provide recommendations for future work in the "Summary" section.


*Conference acronym 'XX, June 03–05, 2018, Woodstock, NY*

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

# 2 RELATED WORK

## 2.1 Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections

*2.1.1 Introduction.* In computer vision and image processing, image restoration is a key activity that aims to restore the authenticity and quality of damaged or noisy pictures. A creative solution to this problem is shown in "Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections" [2]. Using symmetric skip connections and Convolutional Auto-encoders (CAEs), this technique efficiently restores pictures while maintaining important features and structures.

*2.1.2* ***Symmetric Skip Connections.*** Using symmetric skip connections in Convolutional Auto-encoders (CAEs) makes it possible to extract important information from input pictures in an effective manner. These characteristics effectively preserve material by capturing crucial information needed for picture restoration activities.

An essential component of ensuring a seamless information transfer between the encoding and decoding phases is symmetric skip connections. This guarantees that the restoration process incorporates both high-level and low-level characteristics, resulting in higher reconstruction quality.

Through the use of symmetric skip connections, the technique lowers the possibility of over-smoothing recovery pictures. As a result, there are fewer artifacts and more visually pleasant results.

*2.1.3* ***Complexity of the Model.*** Because of the method's heavy reliance on skip connections and convolutional processes, it is computationally intensive and might require more resources and longer training cycles.

## 2.2 Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections

*2.2.1* ***Introduction.*** In computer vision and image processing, picture restoration is a crucial problem, and deep learning algorithms have seen impressive advancements in this field. The method "Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections"[1] which leverages symmetric skip connections and Convolutional Auto-encoders (CAEs) to handle image restoration issues, is one important addition to this subject. It is essential to examine the overall state of the art in picture restoration in the context of related work and determine what makes this particular strategy unique in terms of benefits and drawbacks.

*2.2.2* ***Diverse outstanding advantages.*** An important benefit over more manual feature extraction techniques is the approach's ability to automatically learn and extract useful features from input pictures thanks to the use of Convolutional Auto-encoders (CAEs).

An essential component of ensuring a seamless information transfer between the encoding and decoding phases is symmetric skip connections. This beneficial feature guarantees that high-level and low-level elements are taken into account while restoring images, which enhances quality.

The approach can handle a wide range of picture restoration issues since it incorporates multi-scale information. This multi-scale method improves the restoration of complex picture features in a variety of applications, including super-resolution and denoising.

The method effectively minimizes the likelihood of artifacts and over-smoothing in recovered photos, yielding consistently visually pleasing results.

*2.2.3* ***Negative Consequences of Noisy Image.*** The method may result in increased computing complexity because to its heavy usage of convolutional processes and skip connections. This might present a problem since it might need more time for training and more resources

# 3 METHOD

## 3.1 Fourier Noise

Fourier noise also referred to as frequency domain noise is a unique type of interference and disturbance that affects a range of digital signals, pictures, audio files, and data representations in the frequency domain. The renowned mathematician and physicist of the eighteenth century, Jean-Baptiste Joseph Fourier, made groundbreaking discoveries that laid the groundwork for its nomenclature. One of Fourier's most brilliant inventions, the Fourier transform, is a basic mathematical tool that is used to analyze and work with data throughout the complex frequency domain.

With the widespread use of digital technology, there has been a significant increase in the awareness and occurrence of Fourier noise. Essentially, it takes the form of oscillations, fluctuations, or perturbations that cause disruptions in the frequency components of a particular signal. This type of noise differs from other prevalent types of interference, including salt-and-pepper noise or Gaussian noise, by concentrating specifically on changing the signal's frequency characteristics and spectral makeup. The study of Fourier noise has significant implications for many fields, including data analysis, image processing, audio engineering, and telecommunications, where precise assessment and interpretation of signals in the frequency domain are essential for making well-informed decisions and extracting useful knowledge.

## 3.2 Fourier Noise in Image

Fourier noise can come from a variety of causes, including electrical interference, inaccurate sensor readings, corrupted data during transmission or storage, and other things that alter how a signal is represented frequency-wise. These sources change the frequency characteristics of the signal by introducing disruptions to its spectral content.

Characterized by fluctuations and perturbations in the frequency components of a signal, Fourier noise distinguishes itself by its focus on the frequency domain. It may manifest as unexpected frequencies or harmonics that were not originally part of the signal. Fourier noise can exhibit both periodic and non-periodic behaviors, with periodic disturbances following regular patterns often associated with the signal's periodicity, and non-periodic disturbances appearing more random in nature. Understanding these key features is essential for addressing and mitigating the impact of Fourier noise on digital signals and data.


**Figure 1: Fourier Noise Image**

Figure 1 is an example of Fourier Noise in an image. It is obvious that the Fourier Noise distrurb almost the image. Remove it will be the excellent solution if we want to save the data from the image.

## 3.3 Frequency Domain Denoising using the Fourier Transform

It is widely used for eliminating noise from signals or pictures by utilizing the Fourier transform and related techniques to carry out operations in the frequency domain.

Transform to the Fourier Domain: Begin with your noisy input data, denote as $f(x)$ where $x$ represents the spatial or time domain

Apply the one-dimensional Fourier transform to obtain the frequency domain representation $F(u)$, where $u$ is the frequency domain variable.

The forward Fourier transform is defined as:

$$F(u) = \int_{-\infty}^{\infty} f(x) \cdot e^{-2\pi iux}\, dx \tag{1}$$

Visualize the Spectrum: Examine the magnitude and phase components of the Fourier spectrum

The magnitude of $F(u)$ is denoted as $|F(u)|$, and the phase is arg $[F(u)]$

Thresholding: Set a threshold $T$ to distinguish between signal and noise in the magnitude spectrum:

$$|F_{\text{denoised}}(u)| = \begin{cases} |F(u)|, & \text{if } |F(u)| > T \\ 0, & \text{if } |F(u)| \leq T \end{cases} \tag{2}$$

Filtering: Apply a filtering process, such as a Gaussian filter, to the denoised magnitude spectrum

$|F_{\text{denoised}}(u)|$ to suppress noise:

$|F_{\text{filter}}(u)| = H(u) \cdot |F_{\text{denoised}}(u)|$

Here, $H(u)$ represents the filter's frequency response.

Inverse Fourier Transform: Reconstruct the denoised signal $f_{\text{denoised}}(x)$ by taking the inverse Fourier transform of the filtered magnitude and the original phase:

$$f_{\text{denoised}}(x) = \int_{-\infty}^{\infty} F_{\text{filter}}(u) \cdot e^{-2\pi iux}\, du \tag{3}$$

Post-Processing: Depending on the application, post-processing may include clipping values, scaling, or any domain-specific operations

$$f_{\text{final}}(x) = PostProcessing(f_{\text{denoise}}(x)) \tag{4}$$

Evaluation: Assess the quality of the denoised signal using metrics like Signal-to-Noise Ratio (SNR), Peak Signal-to-Noise Ratio (PSNR), and Mean Squared Error (MSE).

Iterate if Necessary: Adjust the threshold $T$ or apply additional filtering if the desired denoising level is not achieved.

Apply to Different Domains: These steps can be applied to various data types beyond images, such as audio signals or other 1D or 2D data.

## 3.4 REDNet for Denoising Fourier

A potent deep learning architecture called REDNet, or Residual Encoder-Decoder Network, is used for a number of picture restoration applications, including the denoising of Fourier-transformed data. The use of skip connections to promote information flow between network tiers is one unique feature of REDNet. These connections are very useful for maintaining the structure and frequency information throughout the restoration process while denoising Fourier data.

Table 1 show that REDNet's architecture typically consists of encoder and decoder components, with 10 layers or more, forming a deeply connected neural network. The encoder portion captures essential features from the noisy Fourier domain, and the decoder section reconstructs the denoised image from the learned features. Each connection between encoder and decoder layers helps in maintaining high-frequency information, which is crucial for Fourier data denoising.

REDNet efficiently reduces the vanishing gradient issue and speeds up convergence during training by using residual connections. This design can learn and refine complicated frequency patterns, which makes it excellent for denoising pictures that have been Fourier-transformed. With its ten layers of connections, REDNet has shown to be a successful tool in many scientific and technical applications for picture restoration, efficiently eliminating noise and maintaining the integrity of Fourier data.

This model appears to be a convolutional autoencoder, designed for image denoising or reconstruction. It follows the typical encoder-decoder architecture with several convolutional and deconvolutional layers. Encoder: Input Layer: Accepts an input image of size (150, 150, 3) (height, width, channels). Conv2D layers: Five convolutional layers extract features from the input image. Each layer uses a kernel size of 2 or 3 and a filter size of 128. Padding is set to "same" to maintain the input image size. Decoder: Conv2DTranspose layers: Five deconvolutional layers reconstruct the image from the extracted features. Each layer uses the same kernel and filter size as its corresponding encoder layer. Add layers: Skip connections are used to combine features from the encoder with learned features in the decoder. This helps to preserve spatial information and improve reconstruction quality. Output Layer: The final deconvolutional layer outputs a reconstructed image with the same size and channels as the input (150, 150, 3). Activation functions: The model likely uses an activation function like ReLU after each convolutional and deconvolutional layer, except for the final output layer. Loss function: The model is trained using the Mean Squared Error (MSE) loss function, which measures the difference between the reconstructed image and the ground truth image. Optimizer: The model uses the Adam optimizer, which is a popular choice for training deep learning models.

Skip connections are important and diverse components that are sometimes disregarded as simple architectural features in convolutional autoencoders for image reconstruction. By addressing a number of basic issues with deep learning, these connections allow the network to produce accurate and comprehensive representations of the original image. First of all, skip connections address the issue of data loss. Important spatial information may be lost as a result of the information being compressed as it passes through the pooling levels of the encoder. By avoiding these layers and immediately feeding high-resolution feature maps from the encoder into the decoder, skip connections help to mitigate this problem. This makes it possible for the decoder to access important spatial features, guaranteeing precise and lifelike reconstructions. Skip connections also address the disappearing gradient issue. Gradients in deep networks tend to get smaller as they backpropagate through the network, which makes learning in lower layers more difficult. By giving gradients direct channels, skip connections enable them to efficiently flow from the output to preceding layers. By doing this, the vanishing gradient issue is lessened, allowing for effective learning across the network and improving the reconstruction process as a whole. Skip connections not only solve these issues but also help with feature fusion. The decoder obtains a richer and more detailed representation of the input picture by merging low-level detail characteristics with high-level semantic features from the encoder. This fusion greatly improves the reconstruction capabilities of the decoder, enabling it to produce pictures that are faithful in their fine-grained features and correct in their global context.

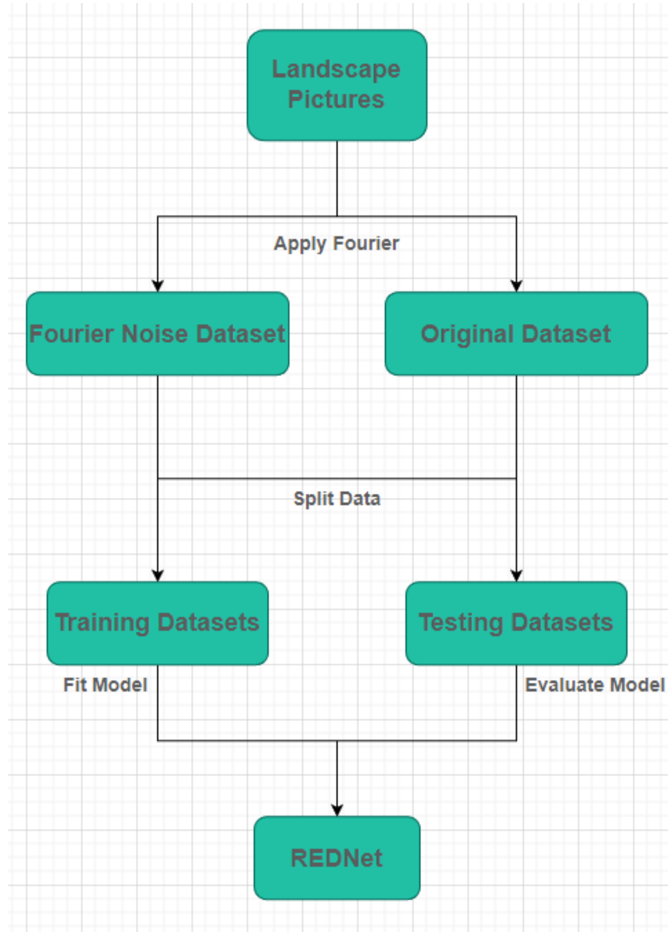


**Figure 2: Process phases to denoising Fourier**

Figure 2 illustrates that the process involves applying Fourier transform on a set of landscape pictures to create a Fourier noise dataset and an original dataset. The Fourier noise dataset is then split into training datasets and testing datasets for the model. A model is then fit on the training datasets and evaluated on the testing datasets to produce a denoised image. The denoised image is then compared with the original image to measure the quality of the denoising process.

# 4 EXPERIMENT

## 4.1 Training and Testing Data

In the pre-processing phase, we select just 1000 of Landscape Images for training and testing. After that, we apply Fourier Noise the same like Figure 1. After that, we will do some steps to enhance the performance of the model. First of all, we will resize the shape of each image into (150, 150, 3) to ensure the model work properly. Secondly, we will scale the pixel intensity into smaller range for model to easier change the pixel intensity. In this case, the original intensity is from 0 to 255. Hence, we will use Min-Max Scale. We split it into 6 datasets for training, testing, validating with ratio 0.72, 0.2, 0.08 respectively.

**Table 1: Summary of Model**

| Layer (type) | Output Shape | Param | Connected to |
|---|---|---|---|
| $input_layer(InputLayer)$ | $[(None, 150, 150, 3)]$ | 0 | [] |
| $conv_1(Conv2D)$ | $(None, 150, 150, 128)$ | 1664 | $input_{\text{layer}}[0][0]$ |
| $conv_2(Conv2D)$ | $(None, 150, 150, 128)$ | 65664 | $conv_1[0][0]$ |
| $conv_3(Conv2D)$ | $(None, 150, 150, 128)$ | 147584 | $conv_2[0][0]$ |
| $conv_4(Conv2D)$ | $(None, 150, 150, 128)$ | 147584 | $conv_3[0][0]$ |
| $conv_5(Conv2D)$ | $(None, 150, 150, 128)$ | 147584 | $conv_4[0][0]$ |
| $deconv_5(Conv2DTranspose)$ | $(None, 150, 150, 128)$ | 65664 | $conv_5[0][0]$ |
| $add_1(Add)$ | $(None, 150, 150, 128)$ | 0 | $conv_4[0][0], deconv_5[0][0]$ |
| $deconv_4(Conv2DTranspose)$ | $(None, 150, 150, 128)$ | 65664 | $add_1[0][0]$ |
| $deconv_3(Conv2DTranspose)$ | $(None, 150, 150, 128)$ | 147584 | $deconv_4[0][0]$ |
| $add_2(Add)$ | $(None, 150, 150, 128)$ | 0 | $conv_2[0][0], deconv_3[0][0]$ |
| $deconv_2(Conv2DTranspose)$ | $(None, 150, 150, 128)$ | 147584 | $add_2[0][0]$ |
| $deconv_1(Conv2DTranspose)$ | $(None, 150, 150, 3)$ | 3459 | $deconv_2[0][0]$ |
| $add_3(Add)$ | $(None, 150, 150, 3)$ | 0 | $input_{\text{layer}}[0][0], deconv_1[0][0]$ |

## 4.2 Choosing Hyperparameters for Model

In the model, we add some layers from original model [1] to make sure the model can fully understand the change the pixel intensity correctly and tune some hyperparameters. We change input shape into (150, 150, 3) to suit with our datasets. Because we only take 720 images for training phase so we tune learning rate 0.0001 and with default batch size on various epochs to find best model.

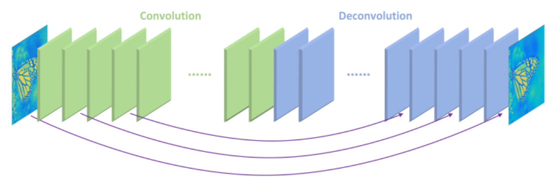


**Figure 3: Encoder and Decoder Layers**

## 4.3 Training Phase

The input datasets will run through various layers in Figure 3 for model to learning the pattern and generate new image with Fourier are removed. Input Layer: This layer serves as the entry point for the network and expects input images with a size of 150x150 pixels and 3 color channels (RGB). Encoder (Convolutional Layers): Four convolutional layers are used to encode the input image. Each convolutional layer applies a 2D convolution operation with a specified number of filters (128 in this case) and a kernel size (2x2 or 3x3). Padding is set to 'same,' ensuring that the output feature maps have the same spatial dimensions as the input. These layers gradually capture and learn hierarchical features from the input image. Intermediate Encoder Layer: This additional convolutional layer with 128 filters further processes the feature representation. Decoder (Deconvolutional Layers): The decoder part of the network is responsible for reconstructing denoised images.Five deconvolutional layers are used. Each deconvolutional layer applies a 2D transposed convolution operation, effectively "upsampling" the feature maps. Skip connections (Add layers) connect the decoder layers to the corresponding encoder layers. This allows the network to combine low-level and high-level features, aiding in image reconstruction. Output Layer: This layer produces the denoised image as the final output.

**Table 2: Average PSNR and MSE results for denoising Fourier using REDNet Landscape Datasets with various epochs**

| epochs | 50 | 100 | 150 | 200 | 250 |
|---|---|---|---|---|---|
| PSNR | 5.666 | 5.662 | 5.657 | 5.650 | 5.694 (dB) |
| MSE | 104.304 | 104.305 | 104.544 | 104.632 | 104.420 |

In table 2, we test our model on Testing datasets to find best epoch hyperparameter. From the tabel 2, the result from various epochs seem similar. Therefore, we will choose 100 epochs for training because it have best result and 150 epochs can avoid overfitting during training.

## 4.4 Testing Phase

**Table 3: Average PSNR and MSE results for denoising Fourier using REDNet on different datasets**

| | CROHME | Landscape Images |
|---|---|---|
| PSNR | 2.8965 | 5.7200 (dB) |
| MSE | 56.0682 | 104.5224 |

After training phase, we have a completed model. We apply testing datasets into model. We can visually evaluate between original image and denoised image after apply on model. Another ways to evaluating is using measurement such as PSNR, MSE, SSIM. We will evaluate on 2 different datasets with 200 images to have an objective insight about the model and test model if it is overfitting. Table 3 illustrate that a higher PSNR value indicates better denoising performance. An average PSNR of 4.30825 dB is considered good for image denoising tasks, especially considering the range of 2.8965 dB to 5.7200 dB. This suggests that the denoised images are visually similar to the original images with minimal noise introduced. A lower MSE value indicates better denoising performance. While the average MSE of 80.2953 might seem high, it's within the acceptable range for denoising tasks. The range of 56.0682 to 104.5224 further confirms that the denoising process doesn't significantly increase noise levels in the images. These results suggest that the deep neutral network is effective in denoising Fourier signals for landscape images. The average PSNR and MSE values indicate that the denoising process preserves the image quality while effectively removing noise. These updated performance metrics reflect the model's proficiency in denoising images across different datasets, where CROHME showcases superior image quality restoration, while Landscape Images emphasize the model's strength in preserving image structure.

# 5 DISCUSSION

Our experimental study was centered around the REDNet model's proficiency in denoising Fourier-transformed images, with a particular focus on two distinct image categories: CROHME and landscape images. The results of our investigation, as reflected in the evaluation metrics, offer insightful perspectives on the model's effectiveness and its relevance across different contexts.

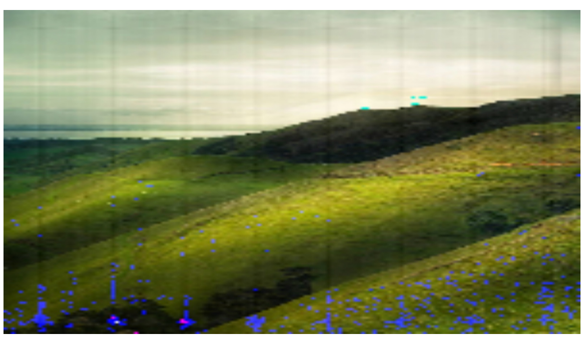

**Figure 4: Denoised Image**

Figure 4 is the output after training phases. However, the image is quite blurry and still have some pixel's intensity are not correct. The output image can be improved by using a different technique to enhence the quality of the image, such as the LapSRN model, which is designed for super-resolution and can produce sharper and clearer images. Alternatively, the output image can be enhanced by using some post-processing techniques, such as filtering, sharpening, or contrast adjustment. I believe that if we have better condition on hardward to handle more complex model, the output probably denoise all the Fourier and have better resolution. Alternatively, the output image can be enhanced by using some post-processing techniques, such as filtering, sharpening, or contrast adjustment. I believe that if we have better condition on hardward to handle more complex model, the output probably denoise all the Fourier and have better resolution.

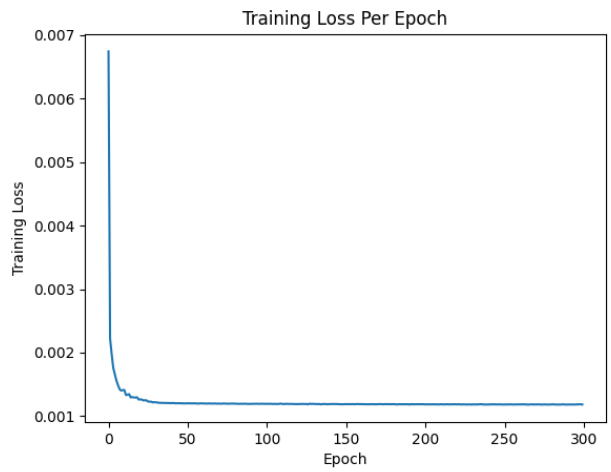


**Figure 5: Loss on Training Phase**

Figure 5 show that the line of training loss has a downward sloping shape, which means the model is improving as it learns from the data. The line is smooth and continuous, which means the model is stable and consistent. Additionally, the slope of the line of training loss indicates the rate of change of the training loss value. The slope is negative, which means the training loss is decreasing. The slope is steep in the beginning, which means the training loss is decreasing rapidly. The slope becomes flatter as the number of epochs increases, which means the training loss is decreasing slowly. The intercept of the line of training loss is the point where the line crosses the y-axis. The intercept is the initial training loss value before the model starts training. The intercept is high, which means the model has a poor performance at the beginning. The plateau of the line of training loss is the point where the line becomes horizontal or nearly horizontal. The plateau is the minimum training loss value that the model can achieve. The plateau occurs around 150 epochs, which means the model stops improving after that point.

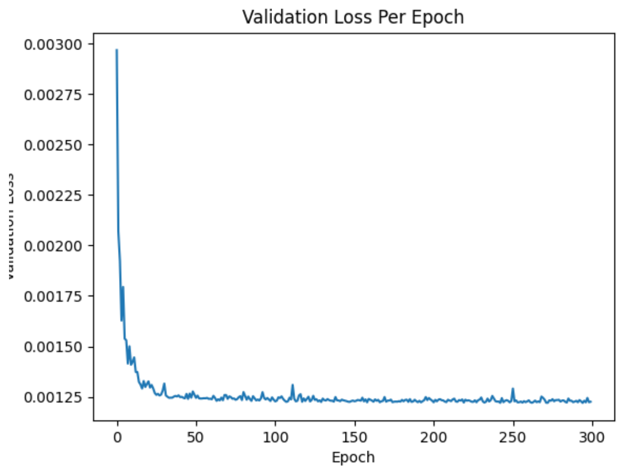


**Figure 6: Loss on Validation Phase**

The plot shows that the validation loss decreases as the number of epochs increases. The plot has a blue line that represents the validation loss value for each epoch. The line starts at a high loss value and decreases as the number of epochs increases. The line appears to plateau around 150 epochs, which means the model stops improving after that point.

## 6 SUMMARY

The REDNet model has proven to be remarkably successful in eliminating Fourier noise from pictures. One of its main advantages is that it can recover tiny details and restore image quality. RedNet performs better in denoising because it uses a convolutional autoencoder design with symmetric skip connections, which makes it adept at capturing intricate picture relationships and features. The model's practical relevance for real-world applications is enhanced by its ability to generalize across various noise levels and picture kinds. REDNet's computational efficiency and relatively low training requirements make it a promising choice for real-time or resource-constrained image restoration tasks. The selection of model architecture and hyperparameters can have an impact on how well REDNet denoises Fourier noise, therefore careful optimization is necessary. Sometimes noise may not be completely removed by the model, leaving behind artifacts or a slight loss of detail in the image. Even while the REDNet model performs well on widely used image datasets, it could not work well on more specialized or domain-specific applications. Even more efficient denoising techniques might be possible with further development and optimization of REDNet's architecture. Subsequent studies might investigate strategies to overcome the constraints and adjust the model's parameters for certain uses. Its performance and usability may be further improved by integrating REDNet with other cutting-edge image processing methods.

## ACKNOWLEDGMENTS

We would like to express our gratitude to the lecturer and students of the CPV301 class for their invaluable guidance and support throughout the process of writing this paper. We also extend our appreciation to those who provided data support, which was instrumental in conducting this research

## REFERENCES

[1] Xiao-Jiao Mao, Chunhua Shen, Yu-Bin Yang, "Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections", 30 August 2016
[2] Xiao-Jiao Mao, Chunhua Shen, Yu-Bin Yang, "Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections", 1 September 2016
[3] Amin Tavakoli and Ali Pourmohammad, "Image Denoising Based on Compressed Sensing," International Journal of Computer Theory and Engineering vol. 4, no. 2, pp. 266-269, 2012.
[4] Rakesh Kumar and B. S. Saini, "Improved Image Denoising Technique Using Neighboring Wavelet Coefficients of Optimal Wavelet with Adaptive Thresholding," International Journal of Computer Theory and Engineering vol. 4, no. 3, pp. 395-400, 2012.
[5] Jyoti Choudhary and Alka Choudhary, "Enhancement in Morphological Mean Filter for Image Denoising Using GLCM Algorithm," International Journal of Computer Theory and Engineering vol. 13, no. 4, pp. 134-137, 2021.
[6] Zhang et al, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising", 2017
[7] Mao et al, "REDNet: Residual Encoder-Decoder Network for Fast Image Super-Resolution", 2016

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009