Genetic Algorithm Based Deep Multi-Route Self-Attention for Single Image Super-Resolution

Miss Nisawan Ngambenjavichaikul

จีเนติกอัลกอริทึมแบบเน้นความสนใจในตัวหลายเส้นทางเชิงลึกสำหรับภาพความละเอียดสูงยวดยิ่งแบบภาพเดียว

น.ส.นิศาวรรณ งามเบญจวิชัยกุล

Thesis Title   Genetic Algorithm Based Deep Multi-Route Self-Attention for Single Image Super-Resolution
By   Miss Nisawan Ngambenjavichaikul
Field of Study   Electrical Engineering
Thesis Advisor   Associate Professor SUPAVADEE ARAMVITH, Ph.D.

Accepted by the FACULTY OF ENGINEERING, Chulalongkorn University in Partial Fulfillment of the Requirement for the Master of Engineering

.... Dean of the FACULTY OF ENGINEERING
(Professor SUPOT TEACHAVORASINSKUN, D.Eng.)

THESIS COMMITTEE

.... Chairman
(Assistant Professor Thavida Maneewarn, Ph.D.)

.... Thesis Advisor
(Associate Professor SUPAVADEE ARAMVITH, Ph.D.)

.... Examiner
(Associate Professor CHARNCHAI PLUEMPITIWIRIYAWEJ, Ph.D.)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

นิศาวรรณ งามเบญจวิชัยกุล : จีเนติกอัลกอริทึมแบบเน้นความสนใจในตัวหลายเส้นทางเชิงลึก สำหรับภาพความละเอียดสูงยวดยิ่งแบบภาพเดียว. (Genetic Algorithm Based Deep Multi-Route Self-Attention for Single Image Super-Resolution) อ.ที่ปรึกษาหลัก : รศ. ดร.สุภาวดี อร่ามวิทย์

การสร้างคืนภาพความละเอียดสูงยวดยิ่งแบบภาพเดียว คือกระบวนการที่ตั้งใจจะสร้างภาพความละเอียดสูงขึ้นมาใหม่จากอินพุตที่มีความละเอียดต่ำจำนวนหนึ่งภาพ แม้ปัจจุบันนี้มีงานวิจัยการสร้างคืนภาพความละเอียดสูงยวดยิ่งแบบภาพเดียวโดยใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันนัล แต่โมเดลที่ใช้กลไกเน้นความสนใจในตัวเองหรือแบบทรานสฟอร์มเมอร์นั้นก็เริ่มเป็นที่รู้จักอย่างกว้างขวางและถูกนำมาศึกษา ซึ่งมีประสิทธิภาพที่ดีในการแก้ปัญหางานทางด้านภาพ งานวิจัยอ้างอิงของเราคือการสร้างคืนภาพความละเอียดสูงยวดยิ่งแบบภาพเดียวโดยใช้ซวินทรานสฟอร์มเมอร์ โดยงานดังกล่าวประยุกต์ใช้วิธีการเลื่อนหน้าต่างและเน้นความสนใจในตัวบนหน้าต่างเฉพาะแห่งที่ไม่ซ้อนทับกัน พร้อมหาความสัมพันธ์ข้ามหน้าต่างด้วย อย่างไรก็ตาม การออกแบบอัลกอริทึมนี้ใช้การปรับค่าแบบคงตัว เพื่อให้ได้ค่าเหมาะสม วิทยานิพนธ์นี้นำเสนอจีเนติกอัลกอริทึมแบบเน้นความสนใจในตัวหลายเส้นทางเชิงลึกสำหรับภาพความละเอียดสูงยวดยิ่งแบบภาพเดียวโดยใช้ขั้นตอนวิธีทางพันธุกรรมค้นหาจำนวนฟิลเตอร์และเลเยอร์ที่เหมาะสม ผลการทดลองแสดงให้เห็นว่าประสิทธิภาพของวิธีการที่นำเสนอในการสร้างคืนภาพความละเอียดสูงยวดยิ่งมีคุณภาพในเชิงค่าพีเอสเอ็นอาร์ได้สูงสุดถึง 0.14 เดซิเบล และ 0.06 เดซิเบลโดยเฉลี่ย กับฐานข้อมูลทดสอบเมื่อเปรียบเทียบกับวิธีการที่ทันสมัยในปัจจุบัน

สาขาวิชา	วิศวกรรมไฟฟ้า	ลายมือชื่อ
		นิสิต ....
ปีการศึกษา	2565	ลายมือชื่อ อ.ที่ปรึกษา
		หลัก ....

6372066821 : MAJOR ELECTRICAL ENGINEERING
KEYWOR   Single Image Super-Resolution, Transformer, Optimization,
D: Evolutionary Algorithms, Genetic Algorithm
            Nisawan Ngambenjavichaikul : Genetic Algorithm Based Deep Multi-Route Self-Attention for Single Image Super-Resolution. Advisor: Assoc. Prof. SUPAVADEE ARAMVITH, Ph.D.

Image restoration, such as single image super-resolution (SISR), is a long-established low-level vision issue that intends to regenerate high-resolution (HR) images from low-resolution (LR) input counterparts. While state-of-the-art image super-resolution models are based on the well-known convolutional neural network (CNN), many self-attention-based or transformer-based experiment attempts have been conducted. They have shown promising performance on vision problems. A powerful baseline model based on the swin transformer adopts the shifted window approach. It enhances the capability by restricting the model to compute the self-attention function only on non-superimpose local windows while enabling cross-window relations. However, the architecture design is manually fixed. Therefore, the results are not achieving optimal performance. This work presents a genetic algorithm-based deep multi-route self-attention network for single image super-resolution (GA-MRSA). The genetic algorithm (GA) is introduced to discover the more suitable number of filters and layers. Experimental results demonstrate that the proposed optimization technique can produce an SR image with a maximum progressive PSNR of 0.14 dB and an average of 0.06 dB in the testing datasets compared to the state-of-the-art.

Field of Study:	Electrical Engineering	Student's Signature	
Academic Year:	2022	Advisor's Signature	

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I
INTRODUCTION

1.1 Motivation and Problem Statement

Recently, multimedia application has gained more interest in daily life, including video streaming, face recognition, image filtering, and machine-to-machine translation. In addition, image processing and computer vision are two core components of multimedia applications, such as image segmentation, image recognition, object detection, anomaly detection, image analysis, etc. Besides, image super-resolution (SR) has been used in those multimedia applications in order to generate image enlargement for the human visual system. Precisely, image resolution is among the most crucial factors that influence how well an image is perceived. The image with higher resolution offers finer details of the scene and constituent object. High-resolution (HR) images are also fatal for many modern devices, e.g., huge electronic visual displays, high-definition television sets, and portable gadgets like smartphones, tablets, cameras, etc.

(a)　　　　　(b)　　　　　(c)　　　　　(d)
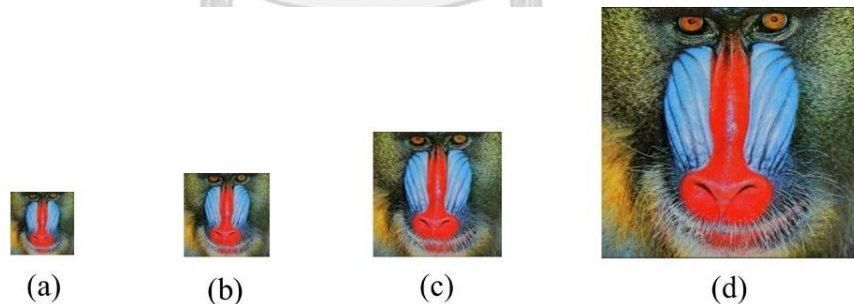
Figure 1: Baboon image in different resolutions

(a) 123×120, (b) 164×160, (c) 246×240, (d) 500×480.

However, low-resolution (LR) images are more frequently gathered in contemporary culture, owing to shortcomings in imaging technology and storage space constraints. Consequently, the techniques for image enlargement are highly sought after. The procedure that gives input as an LR image with sparse details, aiming to reconstruct an HR image counterpart with improved perceptual quality, is referred to as single image super-resolution (SISR). SR models have attained exceptional achievement in many specialties, e.g., medical image processing [1], face [2], security and surveillance imaging [3], remote sensing [4], compressed image and video enhancement [5], object detection [6], and more.

The two primary ways to achieve HR images are hardware-based and software-based approaches. Hardware-based approach's immediate solution to increasing the resolution level is to manufacture sensors with decreased pixel size. The higher-resolution image is thus obtained by raising the pixel density or the number of pixels per unit area. The disadvantage of this method is a reduction in the amount of light coming from each pixel. Shot noise, which significantly lowers the image quality, is produced as the light intensity decreases. Increasing the chip size while retaining the pixel size constant is an alternative way to solve the issue of resolution level escalation. The chip capacitance rises as a consequence of this solution. It is commonly known that a huge capacitance prevents the charge transfer rate from being accelerated. The poor rate of charge transfer severely hampers the creation of images. All hardware solutions to this issue are typically constrained by the price of the necessary image sensors and high-preciseness optics.
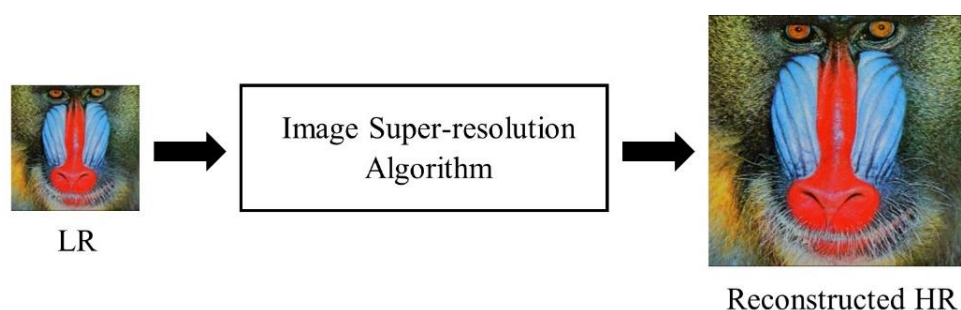
Figure 2: Single Image Super-Resolution

For a software-based technique to increase the resolution level, more precise and quicker algorithms must be designed. The software-based technique is practical because of extensively advanced computation units like the image signal processor (ISP) and graphic processing unit (GPU). The deep learning (DL) technique's computationally demanding goal of attempting to enhance the perceptual quality of LR images has surpassed the traditional techniques. DL is a subset of machine learning (ML) based on straightly learning various data representations. DL-based algorithms intend to automatically discover insightful hierarchical features and utilize them to reach the desired goal. The well-known convolutional neural network (CNN) has long dominated modeling in computer vision, including SISR. It began after AlexNet [7] was introduced to the ImageNet image classification challenge. Its extraordinary capability led to important advancements that greatly affected the discipline as a whole.

Meanwhile, the network structure's development in the sector of natural language processing (NLP) has evolved differently, with the transformer [8] emerging as the preeminent standard. The recurrent and convolution layers are replaced by a straightforward network design that relies mainly on attention mechanisms. Researchers tried extending the transformer to address vision-related issues after experiencing remarkable success in the language area [9, 10]. SwinIR [11] is currently one of the most effective transformer-based vision models. It is a strong backbone model with the basis of swin transformer [12] that employs the shifted window strategy. Increases efficiency by constraining self-attention processing to non-overlapping regional windows while permitting cross-window association. The performance is satisfactory, but the hyperparameter design is still fixed manually. This implies that the hyperparameters can be adjusted to anticipate even better outcomes. Optimization aims to identify the best values among all feasible options. An uncomplicated yet effective optimizer like the genetic algorithm (GA) can be employed to solve this optimization issue.

We introduce a genetic algorithm based deep multi-route self-attention network for single image super-resolution in order to address the optimization problem and improve performance. Utilize the resilience of the genetic algorithm as an optimization technique to optimize the existing self-attention-based model, SwinIR. We are

attempting to search for a more suitable number of filters and layers. In addition, to examine each network design's more suitable hyperparameters.

1.2 Objectives

1. Apply genetic algorithm in single image super-resolution (SISR) in order to improve the quality of reconstructed SR image.
2. Perform super-resolution in different scale factors.
3. Evaluate the performance of the proposed model with the state-of-the-art SISR model.

1.3 Scope of Work

1. Examine genetic algorithm to optimize transformer-based single image super-resolution model.
2. Investigate the proposed method at least on two different scale factors.
3. Assess the performance of the proposed algorithm with the referenced baseline model using subjective and objective image quality assessments.

1.4 Research Procedure

1. Review literature related to single image super-resolution, optimization techniques, and genetic algorithm.
2. Study the methodology of single image super-resolution and select the datasets.
3. Design and develop the architecture of the proposed genetic algorithm network.
4. Train the baseline network with the selected image datasets.
5. Run the proposed genetic algorithm method.
6. Test performance of the proposed method compared with the baseline.
7. Conclude and analyze the experimental results of the proposed algorithm.

1.5 Expected Outcome

1. Achieve a more suitable set of hyperparameters of the referenced baseline model.
2. Reconstruct a better high-resolution output image compared to the original model, in terms of both quantitatively and qualitatively.

1.6 Outline of Thesis

Including chapter I, this thesis consists of a total of five chapters. The rest of the contents are provided with descriptions as follows:

Chapter II: describes the background and literature review related to frameworks and modern techniques of single image super-resolution, such as super-resolution with deep learning and transformer. Moreover, optimization techniques and genetic algorithms in super-resolution are also included.

Chapter III: describes the proposed method that explains the overall structure, chromosome design, fitness function, genetic algorithm operators, and termination criteria.

Chapter IV: demonstrates and analyzes the experimental results compared with the baseline.

Chapter V: comprises of conclusion and future work.

CHAPTER II
BACKGROUND AND LITERATURE REVIEW

2.1 Background

2.1.1 Single Image Super-Resolution

Single Image Super-Resolution (SISR) is a process of reconstructing one corresponding HR image from one LR input image. LR input image I_x is generally modeled as the outcome of the following degradation:

$$I_x = D(I_y; \delta), \tag{2.1}$$

where I_y is the ground truth HR image, D denotes a degradation mapping function and δ is the parameters of the degradation procedure (e.g., noise or scaling factor). However, the degradation procedure (i.e., D and δ) is normally unknown and only LR images are collected. This case also referred to as blind SISR. Given the LR input image I_x, a super-resolved or approximately reconstructed image $\hat{I}_y$ of the ground truth HR image I_y needed to be restored as follows:

$$\hat{I}_y = SR(I_x; \theta), \tag{2.2}$$

where SR is the super-resolution model and θ indicates the parameters of SR.

Even though the degradation procedure is unacquainted and can formed due to many factors (e.g., sensor noise and speckle noise, compression artifacts, anisotropic degradations), researchers are attempting to predict the degradation mapping. The majority of models directly represent the degradation as a single downsampling step, as in:

$$D(I_y; \delta) = (I_x) \downarrow_s, \{s\} \subset \theta, \tag{2.3}$$

where $\downarrow_s$ is a downsampling operation with the scaling factor s. Practically, most generic SR datasets are created according to this pattern, and the most widely used downsampling method is bicubic interpolation with antialiasing.

Presently, SR algorithms can be categorized into three groups based on the approaches employed. They are interpolation-based, reconstruction-based, and machine learning-based SR.

Interpolation-based SISR methods. The most commonly known are nearest neighbor interpolation, bilinear interpolation, and bicubic interpolation. They are straightforward and rapid but have a limited capacity for detail improvement.

Reconstruction-based SISR methods, such as edge sharpening, regularization, and deconvolution. As a matter of fact, these algorithms can generate high-resolution images clear and vivid since they frequently require complicated prior information. Which needs prior knowledge to define constraints for the target HR image. However, many reconstruction-based methods quickly decrease in performance as the scaling factor rises, and these techniques are usually time-consuming.

Learning-based SISR methods. This category of SISR algorithms obtains approximate HR images with the use of machine learning techniques. Due to its fast speed and outstanding performance, the SISR based on machine learning techniques has attracted significant interest. Typically, the approaches employ learning-based algorithms to extract statistical correlations between LRs and their HRs counterparts from a large number of training datasets. Some examples of early learning-based SISR methods are Markov Random Field (MRF), neighborhood embedding, and sparse coding. In addition, a lot of work combines reconstruction and machine learning-based techniques to lessen artifacts brought about by external training data. Recent studies have demonstrated the superiority of SISR algorithms based on deep learning (DL) over previous reconstruction-based and other machine-learning-based methods.

2.1.2 Deep Learning-Based Super-Resolution

Deep learning is a subbranch of the machine learning method based on different representations of direct learning data. When compared to conventional algorithms, deep learning algorithms strive to automatically learn the rich hierarchical representation of the content in order to accomplish the goal; the entire learning process can be viewed as a whole. Deep learning in SR can come in two broad categories: supervised learning and unsupervised learning.
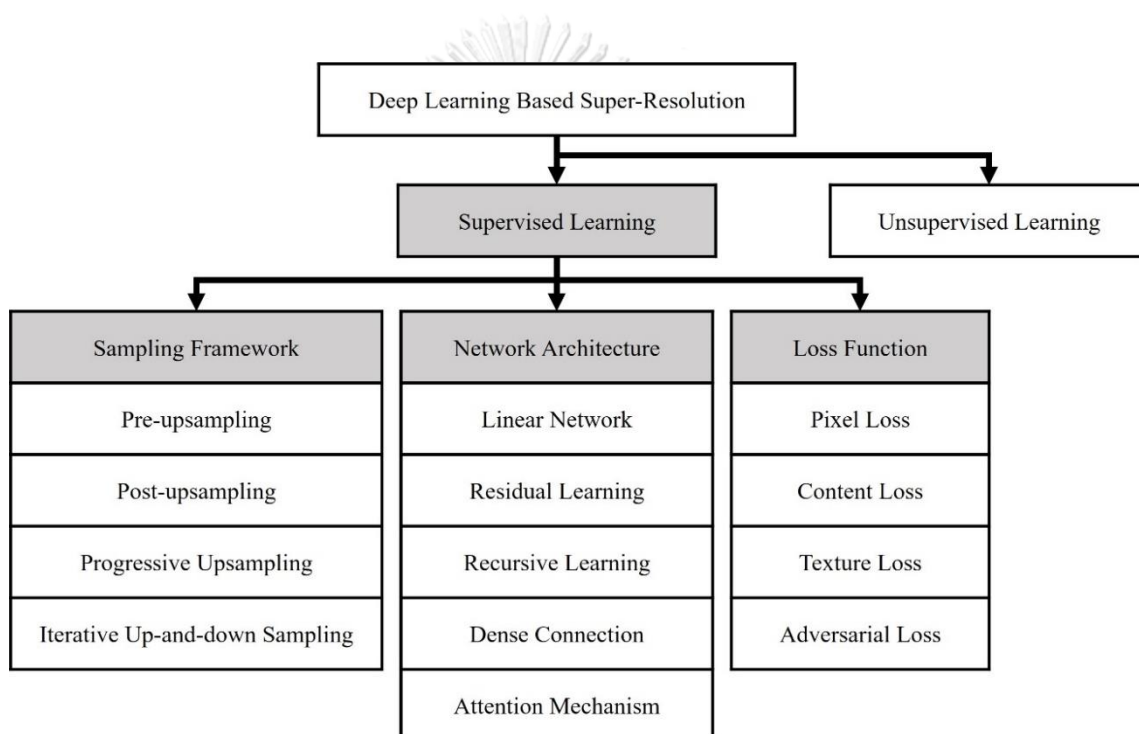
Figure 3: Taxonomy of Supervised Deep Learning-based SR

2.1.2.1 Sampling Framework

SISR is inherently an ill-posed problem since there are multiple solutions that exist for the same LR input image. Thus, the main concern is on how to execute upsampling stage. Even though the frameworks of existing models differ greatly, four model frameworks can be identified corresponding to the upsampling operation used and where they are located inside the model.

- **Pre-upsampling Super-resolution**

Since it is challenging to learn the representation from low-dimensional space to high-dimensional space explicitly, a simple solution is to use conventional upsampling techniques to acquire higher-resolution images and afterward refine them through deep neural networks. Dong et al. [13] introduce SRCNN to generate an end-to-end mapping from interpolated LR images to HR images by first implementing the pre-upsampling SR approach.
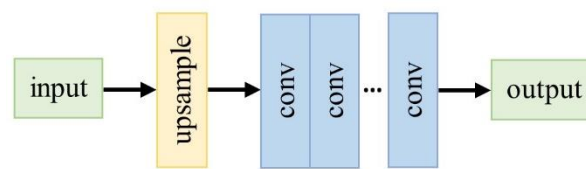
Figure 4: Structure of Pre-upsampling Super-Resolution Framework

To be specific, the LR images are upsampled using interpolation-based methods to obtain HR images with coarse features at the aspire image size. Following that, reconstructed super-resolved images with refined details are produced by employing deep CNNs. Therefore, this framework has steadily risen to be one of the most widely known. Nevertheless, most operations are carried out in high-dimensional space, and negative consequences such as noise or blurring generally occur from the predetermined upsampling. Compared to alternative frameworks, the expense of time and space is substantially higher.

- **Post-upsampling Super-resolution**

With the aim of better computational efficiency and fully utilizing deep learning techniques for image resolution increment, researchers suggest performing the majority of computation in low-dimensional space. It is done by displacing the predetermined upsampling with end-to-end learnable layers merged at the hindmost stage of the model. This concept is recognized as a post-upsampling SR framework.

In the early studies [14, 15], as illustrated in Figure 5, the LR input images are passed through deep CNNs without enhancing resolution, and the end-to-end learnable upsampling layers are applied at the end of the network. Given that the feature extraction task that requires high computational expense takes place in low-dimensional space and the resolution only rises at the very end, the computation and spatial complexity are greatly decreased. Accordingly, this framework has also developed as one of the most widely used frameworks in SR [16-18]. Ensuing models diverge the most with different learnable upsampling layers, DL networks design, and learning tactics.
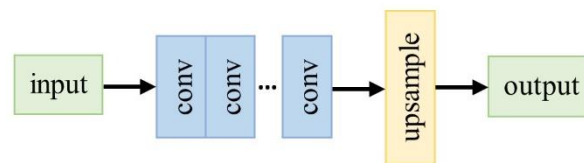
Figure 5: Structure of Post-upsampling Super-Resolution Framework

- **Progressive Upsampling Super-resolution**

Albeit the computational cost of the post-upsampling SR approach has been significantly decreased, it still has a number of drawbacks. Since upsampling is done at a single pace, it makes the learning procedure become more challenging for huge scaling factors. Moreover, it is not suitable for a multi-scale SR model due to the need to train an individual model for every scaling factor. To deal with these limitations, Laplacian pyramid SR network (LapSRN) [19] introduces a progressive upsampling framework. The architecture is presented in Figure 6. In particular, the models implementing this framework gradually generate higher-resolution images based on a succession of CNNs. The images are improved by CNNs and upsampled to better resolution at each stage.
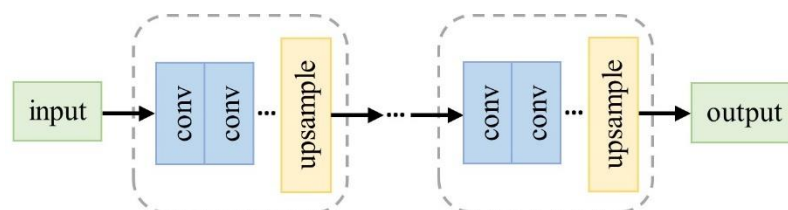
Figure 6: Structure of Progressive Upsampling Super-Resolution Framework

The models under this approach effectively reduce the learning difficulties, large factors for the most part, and also handle the multi-scale SR without adding an excessive amount of spatial and temporal cost. The complicated model design for several stages and the training stability are two issues that these models also run into. Therefore, additional modeling guidance and more sophisticated training techniques are required.

- **Iterative Up-and-down Sampling Super-resolution**

For the purpose of improving the acquisition of the mutual dependencies between LR and HR image pairings, back-projection, an effective iterative process, is integrated into SR. Figure 7 displays the structure of this framework, iterative up-and-down sampling. It aims to recursively use back-projection refinement by evaluating the reconstruction error and then combining it back to adjust the HR image intensity. Based on the idea, a proposal by Haris et al. [20] for DBPN makes use of iterative up-and-down sampling layers. The model reproduces the eventual HR outcome incorporating all of the intermediate reconstructions by interchangeably joining upsampling and downsampling layers. Comparable to this, the SRFBN [21] uses a more compact skip connection iterative up-and-down sampling feedback block to learn stronger representations.
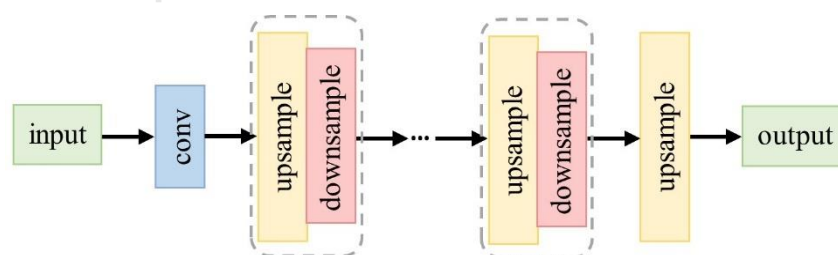
Figure 7: Structure of Iterative Up-and-down Sampling Super-Resolution Framework

However, the back-projection modules' design considerations remain ambiguous. Since this approach is relatively new to deep learning-based SR, additional research is necessary to understand its capabilities fully.

2.1.2.2 Network Architecture

Network architecture is one of the most crucial components of deep learning these days. Scholars in the super-resolution discipline build the final networks utilizing a variety of network design methodologies on the basis of the four SR frameworks. In the following part, these networks are broken down into the fundamental concepts or tactics for network design, explained, and discussed their benefits and drawbacks individually.

- **Linear Network**

Linear networks have a straightforward architecture that includes just one input movement track, no skip connections, and no multiple branches. SRCNN [22] is an example of a linear network. Multiple convolutional layers are piled on top of one another in such network topologies, and the input passes sequentially from the first to the last layers. Different linear networks vary by the location the upsampling operation is executed, e.g., pre-upsampling or post-upsampling scheme.

- **Residual Learning**

Global residual learning and local residual learning are two broad categories for residual learning network design.

Global Residual Learning: Take into account that SISR is a task that transforms image to image, where the given image feeding in is strongly associated with the desired output image. Thus, the researcher strives to comprehend just the residuals among the two images. This network architecture is known as global residual learning. In this instance, it merely necessitates on training a residual map to restore the omitted high-frequency information rather than learning a complex translation from an entire image to another. The model complication and learning difficulties are significantly reduced because the residuals are barely detectable in vast areas. That being so, SR models frequently employ it [23-25].

Local Residual Learning: Inspired by the deep residual learning for image recognition (ResNet) [26], local residual learning is employed to mitigate the degradation problem brought on by continuously expanding network depths, lessen the

complexity of training, and enhance learning capacity. It is also popular for SR models [27, 28].

The difference between the two implementation methods is that the first forthrightly links the input and output images, whereas the second typically escalates multifarious shortcuts between layers of varying depths within the network. Practically, the aforementioned designs are both actualized by skip connections and element-wise summation.
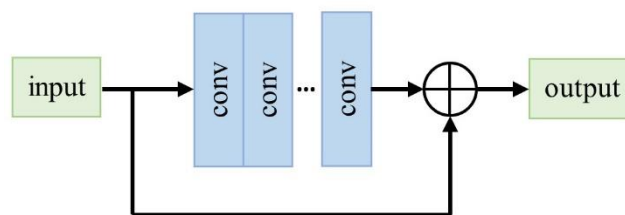
Figure 8: Residual Learning Network Architecture

- **Recursive Learning**

Recursive learning, which involves adopting the identical modules numerous times in a recursive fashion, is proposed in SR with the goal to acquire higher-level characteristics without presenting overpowering parameters. DRCN [29] with 16 recursive layers, yields a receptive field of 41×41 by utilizing a single convolutional layer as the recursive unit. The DRRN [25] achieves even greater performance than the 17-ResBlock baseline by using a ResBlock [26] as the recursive unit for 25 recursions. More recently, Li et al. [21] presented a feedback network based on recursive learning and used an iterative up-and-down sampling SR approach. In this network, the whole network's weights are allocated covering all recursions.
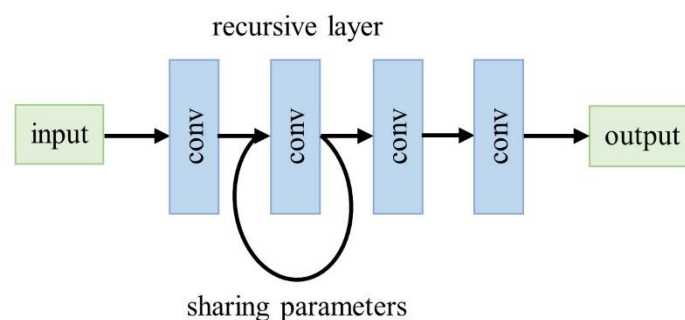
Figure 9: Recursive Learning Network Architecture

Recursive learning can generally learn increasingly complex characterizations without introducing a large number of extraneous parameters, but it still cannot present considerable computational costs. Additionally, it has intrinsic difficulties with inflating or vanishing gradients.

- **Dense Connection**

Dense connection gradually became more famous in vision applications ever since Huang et al. [30] proposed DenseNet based on dense blocks. Each layer in a dense block receives input from all layers that came before it, and its own feature maps are fed as inputs into all successive layers. By using a small growth rate and compressing channels after appending all of the input feature maps, the dense connections significantly decrease the model size while also relieving gradient vanishing, strengthening signal propagation, and inspiring the reuse of features. Dense connections are inserted into the SR field with the purpose of combining low-level and high-level features to offer greater characteristics for restoring high-quality features.
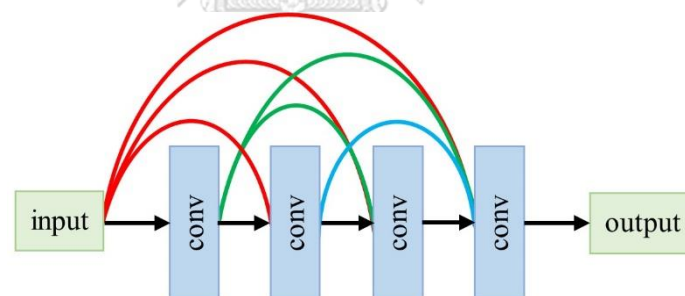
Figure 10: Dense Connection Network Architecture

- **Attention Mechanism**

Channel Attention: In order to increase learning capacity by expressly modeling channel interdependence, Hu et al. [31] present a "squeeze-and-excitation" block. The block considers the interdependence and interaction of the feature representations between various channels. Global average pooling (GAP) is used in the block to compress each input channel into a channel descriptor, which is then supplied into two dense layers to provide channel-wise scaling factors for input channels.

Consequently, Zhang et al. [27] amalgamate the channel attention mechanism with SR and propose RCAN, which revolutionized the model correlation capability and SR efficiency. A second-order channel attention (SOCA) module is subsequently introduced by Dai et al. [32] with the intend of learning the characteristic representations better. The SOCA allows for the extraction of more illuminating and discriminative representations by dynamically rescaling the channel-wise features using second-order feature statistics rather than GAP.

Non-local Attention: The preponderance of SR models in present use has relatively small local receptive fields. Nonetheless, some far-off objects or textures could be essential for creating local patches. To pull out features that preserve the long-range dependencies across pixels, Zhang et al. [33] present local and non-local attention blocks. They particularly imply a trunk branch for feature extraction and a (non-)local mask branch for dynamically rescaling trunk branch features. The non-local branch uses the embedded Gaussian function to assess pairwise associations between each pair of position indices in the feature maps in order to forecast the scaling weights. In contrast, the local branch uses an encoder-decoder architecture to learn the local attention. The proposed method effectively catches spatial attention through this mechanism, strengthening the capacity for representation. The non-local attention mechanism is also employed by Dai et al. [32] to capture long-distance spatial contextual data.

2.1.2.3 Loss Function

Loss functions are employed in the super-resolution discipline to quantify reconstruction error and lead model optimization. Researchers initially used the pixelwise L_2 loss to evaluate reconstruction quality, but they eventually realized that this method was not particularly reliable. To better measure the reconstruction error and provide more accurate and high-quality results, a variety of loss functions, such as adversarial loss [16] and content loss [34], are used. These loss functions have become increasingly significant in recent years.

Pixel Loss. Pixel loss measures the pixel-wise difference between two images. It primarily contains L_1 loss (mean absolute error) and L_2 loss (mean square error):

$$\mathcal{L}_{pixel\_L_1}(\hat{I}, I) = \frac{1}{hwc} \sum_{i,j,k} |\hat{I}_{i,j,k} - I_{i,j,k}|, \qquad (2.4)$$

$$\mathcal{L}_{pixel\_L_2}(\hat{I}, I) = \frac{1}{hwc} \sum_{i,j,k} (\hat{I}_{i,j,k} - I_{i,j,k})^2, \qquad (2.5)$$

where the height, width, and channel number are represented as h, w, and c, respectively. Additionally, another variation of the pixel L_1 loss is called Charbonnier loss [19]., denoted as:

$$\mathcal{L}_{pixel\_Char}(\hat{I}, I) = \frac{1}{hwc} \sum_{i,j,k} \sqrt{(\hat{I}_{i,j,k} - I_{i,j,k})^2 + \epsilon^2}, \qquad (2.6)$$

The reconstructed HR image is constrained by the pixel loss to be sufficiently near to the actual ground truth pixel values. In contrast to L_1 loss, L_2 loss penalizes larger errors but is more forgiving of tiny errors, leading to outcomes that are frequently too smooth. In actual use, L_1 loss outperforms L_2 loss in terms of performance and convergence [17, 35]. The pixel loss progressively emerges as the most used loss function since the definition of PSNR is significantly associated with a pixel-wise difference and decreasing pixel loss directly maximizes PSNR. Nevertheless, because the pixel loss truly ignores image quality, such as textures and perceptual quality, the results frequently lack high-frequency details and have textures that are perceptually unpleasant due to over-smoothing [16, 34, 36].

Content Loss. The content loss is added to SR [34] with the aim of assessing the perceptual quality of the images. Specifically, it makes use of an image classification network that has already been trained to measure semantic distinctions between images. The Euclidean distance between two high-level representations of two images serves as a marker for content loss, given by:

$$\mathcal{L}_{content}(\hat{I}, I; \phi, l) = \frac{1}{h_l w_l c_l} \sqrt{\sum_{i,j,k} \left(\phi_{i,j,k}^{(l)}(\hat{I}) - \phi_{i,j,k}^{(l)}(I) \right)^2} \qquad (2.7)$$

In essence, the content loss conveys the classification network's learned understanding of hierarchical image features to the SR network. The content loss, as opposed to pixel loss, promotes the output image to be perceptually similar to the target image rather than requiring exact pixel matching.

Texture Loss. The texture loss or style reconstruction loss is incorporated into SR due to the need for the reconstructed image to have the same style as the target image (e.g., colors, textures, contrast), which is driven by the style representation [37]. Following [37], the image texture is defined as the Gram matrix and is viewed as the correlations between various feature channels.

$$G_{i,j}^{(l)}(I) = vec\left(\phi_i^{(l)}(I)\right) \cdot vec\left(\phi_j^{(l)}(I)\right) \tag{2.8}$$

$$\mathcal{L}_{pixel\_L_2}(\hat{I}, I) = \frac{1}{hwc} \sum_{i,j,k} \left(\hat{I}_{i,j,k} - I_{i,j,k}\right)^2, \tag{2.9}$$

It provides textures that are far more realistic and results in results that are more pleasing to the eye. Even yet, choosing the patch size to match textures is still an empirical process. Because texture statistics are averaged over regions with various textures, too little or too large of a patch might create artifacts in textured sections or the entire image.

Adversarial Loss. A growing number of vision tasks have been introduced to GANs [38] in recent years due to their strong learning capabilities. To put it more specifically, the GAN is made up of a generator that performs generation such as text generation or image transformation and a discriminator that accepts as inputs the generated results and instances drawn from the target distribution and determines whether or not each input is from the target distribution. In SR, can adopt adversarial loss by treating the SR model as a generator and defining an extra discriminator to judge whether the input is generated or not. Ledig et al. [16] first proposed SRGAN using adversarial loss based on cross entropy as in the equation below. Where images that are randomly sampled from the ground truth denote as I_s . $\mathcal{L}_{gan\_ce\_g}$ and $\mathcal{L}_{gan\_ce\_d}$ represent the adversarial loss of the SR model and the discriminator D, respectively.

$$\mathcal{L}_{gan\_ce\_g}(\hat{I}; D) = -\log D(\hat{I}), \tag{2.8}$$

$$\mathcal{L}_{gan\_ce\_d}(\hat{I}, I_s; D) = -\log D(I_s) - \log\left(1 - D(\hat{I})\right), \tag{2.9}$$

2.1.3 Transformer

In recent years, transformer models [8] have displayed outstanding performance on a variety of language tasks, including machine translation, question answering, and text categorization. BERT (bidirectional encoder representations from transformers) [39], GPT (generative pre-trained transformer) [40], and T5 (text-to-text transfer transformer) [41] are the most well-established models among them. The potential of transformer models to scale up to very large capacity models has made their tremendous impact more obvious. Despite the fact that attention models have been widely employed in both feed-forward and recurrent networks, transformers are based purely on the attention mechanism and have a novel implementation optimized for parallelization, namely multi-headed self-attention. Additionally, transformers are generally pre-trained via pretext tasks on large-scale datasets because they require a less preexisting understanding of the architecture of the topic than convolutional and recurrent. A pre-training like this saves costs by avoiding time-consuming manual annotations and encoding highly expressive and generalizable representations that model complex relationships between the elements in a given dataset. In order to achieve successful outcomes, the learned representations are then improved on the downstream tasks in a supervised way.

The key concept that developed the traditional transformer models is self-attention which discovers the connections among sequence's elements. Transformers enable them to concentrate on whole sequences, which allows them to capture long-range relationships in contrast to recurrent networks, which can only pay attention to short-term context and handle sequence elements iteratively.

❖ **Self-Attention**

Scaled Dot-Product Self-Attention: The transformer uses a scaled dot-product self-attention that functions similarly to the general attention mechanism that existed earlier. The scaled dot-product self-attention, as the title implies, calculates a dot product for every query, q , using all of the keys, k first. After that, it divides each result by $\sqrt{d_k}$ and then applies a softmax activation function. This allows it to acquire the weights needed to scale the values, v. Practically, the scaled dot-product attention's calculations can be applied quickly and effectively to the complete set of queries at once. The attention function is given the matrices Q, K, and V as inputs in order to accomplish this:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \qquad (2.10)$$

The structure of a scaled dot-product self-attention is depicted in Figure 11.
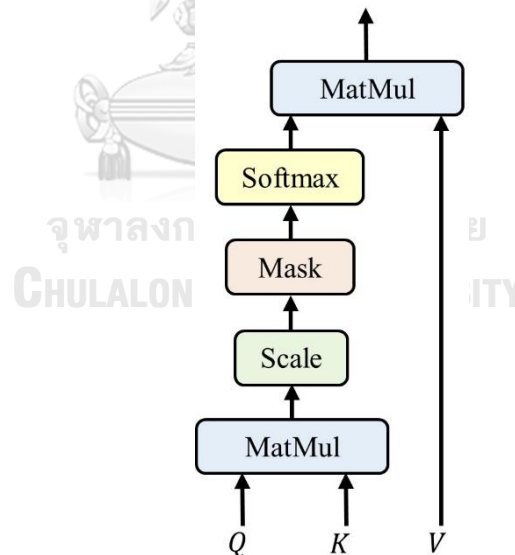
Figure 11: Scaled Dot-Product Self-Attention Function

Multi-Headed Self-Attention: They then proposed a multi-head self-attention mechanism that linearly extends the keys, values, and queries h times, utilizing a unique learned projection each time. Following this, the single self-attention function is parallelly employed to each of the h projections to generate h outcomes, which are

then concatenated and projected once more to yield the final output. The goal of multi-head attention is to enable information extraction from various representational subspaces, which is impossible with a single self-attention head alone. Figure 12 illustrates the structure of a multi-head self-attention mechanism. Which can be defined below:

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O \qquad (2.11)$$

Where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$
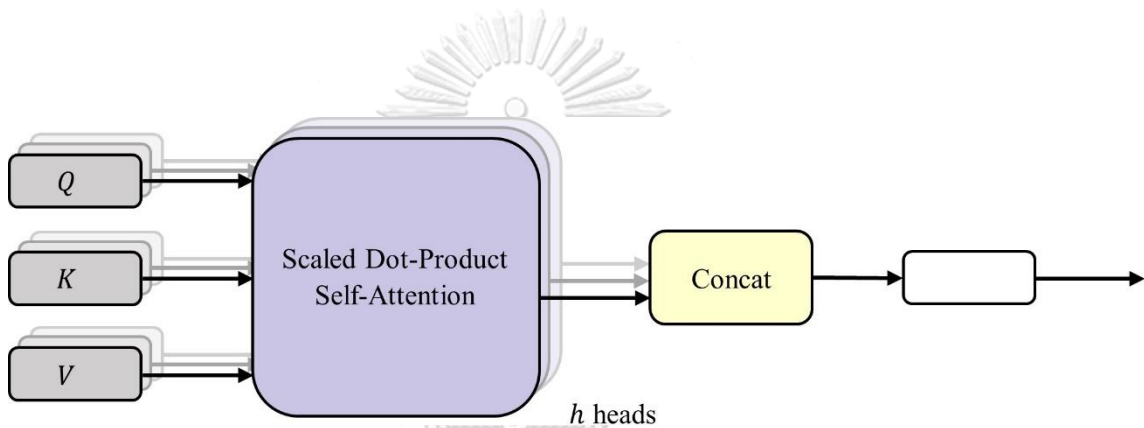
Figure 12: Multi-Headed Self-Attention Function

Self-attention differs from convolution since its filters are dynamically constructed as opposed to convolution's static filters. Furthermore, self-attention is unaffected by input point variations and permutations. Additionally, this could work with unpredictable sources without difficulties unlike normal convolution, which demands a grid system. In essence, self-attention offers the potential to learn both global and local properties, as well as explicitly to understand both the receptive field and kernel weights in an adaptive manner.

2.1.4 Optimization Techniques

2.1.4.1 Overview of Optimization Techniques

When referring to issues in computer science and mathematics, optimization techniques imply a computational context where the objective is to arrive at the best possible choice by repeatedly assessing all reasonable choices. It is a useful method for investigating the suitable performing requirements and the desired design parameters. The optimization approaches utilized to handle the issue are specifically designed to meet the specific issue's nature in order to deliver trustworthy optimal solutions. Technology has progressed to the point that optimization is now a necessary part of many application fields. For example, in the case, a hospital's primary aim would be to minimize the amount of time patients must wait in the emergency unit before being examined by a professional. In this scenario, the assets would comprise doctors, nurses, facilities, tools, and so forth. In business, the intention may be to boost sales by concentrating on the potential consumers while taking practical and financial restrictions into consideration. Without efficient design and operations optimization, production and engineering operations will not be as effective as they are now. It is conceivable to observe how various natural species develop over time and adapt to their surroundings. This comprehension can also inspire human ingenuity. For example, the submersible was motivated by fish, the theory that makes birds fly was used to construct the plane, and radar progressed from bats. Therefore, numerous optimization techniques draw inspiration from nature.
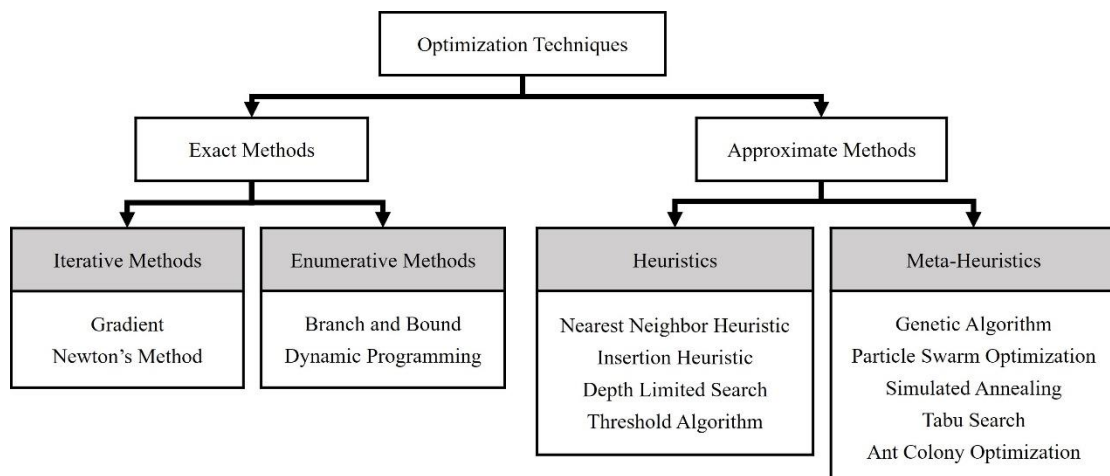
Figure 13: Hierarchy-structured Overview of Optimization Techniques

Before it can be solved, an optimization problem must first be modeled, which entails that it must be expressed mathematically with the use of variables. To discover the most effective answer to these issues across a variety of domains, optimization techniques are needed. Even though small dimension problems can be solved without the aid of computers, larger and more complicated problems often demand specialized techniques and computer simulation. The mathematical summary of the two-step iterative optimization procedure is as follows:

$$x^q = x^{q-1} + \alpha^{\star} S^q \qquad (2.12)$$

The initial stage is to utilize gradient information to determine the search direction S. The next phase is to continue on this path until no more advancement is possible. The optimum step size, $\alpha^{\star}$, is obtained in the second step, which is referred to be a one-dimensional or line search. Note that there are also gradient-based algorithms that do not rely on a one-dimensional search.

There are various ways to classify optimization techniques depending on their purpose and properties. Since there could be a large number of possible configurations and using exact methods to arrive at the optimized solution is not feasible, it is necessary to use intelligent tools to handle these issues. The use of metaheuristic algorithms to address complex optimization issues has been proven to be successful. The appropriate compatibility of these algorithms with many engineering optimization issues is evident.

2.1.4.2 Evolutionary Algorithm

Evolutionary algorithm (EA) [42] is a meta-heuristic optimization method influenced by the evolution of nature organisms. As a method for tackling optimization problems, EAs have drawn a great deal of attention. These techniques have the advantage of being incredibly resilient and are generally inspired by occurrences in nature. They are simple to use, have a higher probability of discovering a global or nearly global optimal, and are well suited for discrete optimization problems. These methods have several significant limitations, including high computational cost, ineffective constraint handling, problem-specific parameter adjustment, and restricted problem size. The evolutionary representation, implementation specifications, and

essence of the particular applicable task vary amongst different types of EA approaches. Two of the most well-known EAs are the more established genetic algorithm (GA) [43] and particle swarm optimization (PSO) [44]. Other algorithms that fall into this category include differential evolution (DE) [45], simulated annealing (SA), tabu search, ant colony optimization (ACO), harmony search (HS), and more. Charles Darwin's theory of evolution serves as the basis for the following three characteristics that define EAs:

Population-based: Once the present answers are inadequate, EAs are employed to optimize the process to provide further effective solutions. The population is the collection of currently employed solutions from which offspring are to be generated.

Fitness-oriented: A fitness value computed from a fitness function is given to each solution. This fitness value assesses the effectiveness of the solution and shows which is preferable.

Variation-driven: Going to consider that none of the existing population's solutions have come close to achieving the stated objective, which corresponds to each chromosome's fitness value. As a consequence, there ought to be an occasion where a new and improved collection of solutions can be established. Thus, new solutions have resulted after some modifications individually.

❖ **Simulated Annealing (SA)**

The inspiration and name of simulated annealing [46] come from the annealing process in metallurgy, a technique that entails heating and controlled cooling of a material to enlarge its crystals' size and lessen their flaws. The slow cooling increases the likelihood that the atoms will find configurations with lower internal energy than the initial one because the heat causes them to become dislodged from their initial positions, a local minimum of internal energy, and wander randomly through states of higher energy. The function to be minimized is regarded as the internal energy of the system in that state in the simulated annealing method, where each point in the search space is compared to a state of some physical system. As a result, the objective is to move the system from any initial state to the state that requires the least amount of energy.

❖ **Particle Swarm Optimization (PSO)**

Particle swarm optimization (PSO) [44] is one of the optimization techniques belonging to EAs. The technique was created by the modeling of streamlined social models. PSO was inspired by the behavior of organisms like fish schooling and bird flocking and integrates socio-cognition human agents with social psychology principles. The findings of the research conducted on a flock of birds indicate that they obtain food by congregating in groups, not individually. PSO simulates behaviors such as a swarm of bees looking for food, for instance. The population, or swarm, converges on the optimal solution utilizing information obtained from both the collective knowledge of the swarm as a whole and from each individual, also mentioned to as a particle. An initial population that is randomly distributed across the design space serves as the starting point of the algorithm. Then, from one design iteration to the next, position of each particle is updated utilizing the equation below:

$$x_i^{q+1} = x_i^q + v_i^q \Delta t \qquad (2.13)$$

where q denotes the qth iteration, i denotes the ith individual, and v_i^q denotes the velocity of ith individual at q^{th} iteration. Δt is the time increment which is generally represented as unity. Each particle is first defined as a random velocity vector that will be updated with Equation (2.14) after every iteration.

$$v_i^{q+1} = wv_i^q + c_1 r_1 \frac{(p_i - x_i^q)}{\Delta t} + c_2 r_2 \frac{(p^g - x_i^q)}{\Delta t} \qquad (2.14)$$

❖ **Genetic Algorithm (GA)**

Genetic algorithm (GA) is one of the most well-known EA methods. Essentially, it was theoretically validated [47] and also exhibited captivating benefits while manipulating a diversity of optimization problems. John Holland from the University of Michigan [48] first presented the concept of a genetic algorithm in 1975. Fig. 2.10 depicts the typical flowchart of GA.

GA can address complex tasks with various variables and a broad assortment of possible outcomes by replicating the evolutionary process of nature to reach a particular goal. Chromosomes are used in GA as a metaphor for the solutions. Each chromosome's fitness value is evaluated using the fitness function, and the values are then sorted from best to worst. In GA, generating new solutions resembles the dynamics of natural selection and genetic inheritance that take place in living creatures. Different nature-inspired operators are repetitively employed in this stage. They are, namely, selection, crossover, and mutation [49].
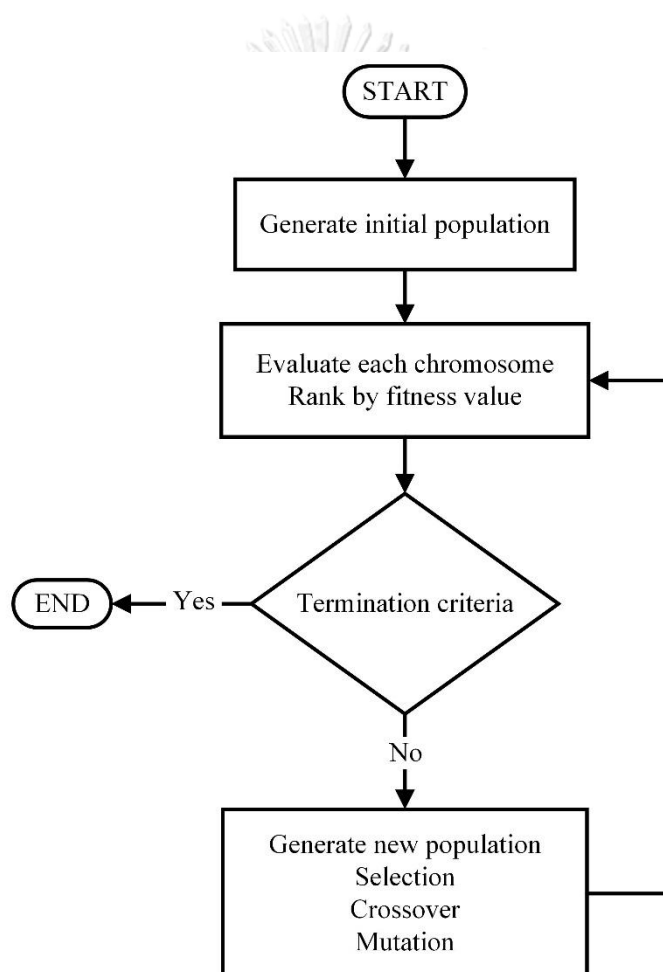
Figure 14: Flowchart of the typical GA

A number of the fittest chromosomes are first selected as parents before producing a new collection of chromosomes. To imitate the survival of the fittest, the chromosomes with higher fitness are chosen more frequently compared to those with

lower fitness. Different selection procedures may call for different ways of probability assignment in order to maintain a favorable balance among the diversification of the new population and the evolution of chromosomes. There have been many selection techniques introduced, but the two that are most commonly applied are the tournament selection and the roulette wheel selection. The crossover operator unites the parent chromosomes after identifying the parent chromosomes to generate new offspring. There are several crossover methods put forth, including multi-point and uniform crossover. After a few generations, it is plausible that the new solutions may not alter much because stronger individuals are being chosen more often, which could lead to population stagnation. A mutation operation is a method used to preserve population variation and avoid stagnation.

On the account of the GA's high degree of parallelism and adaptability, global optimization nature, and ability to handle challenging search spaces even with limited prior knowledge. Hence, this optimization method has found broad use in a variety of domains, including design and scheduling, system control, function optimization, power systems, image processing, etc. There are many GA software tools available from various suppliers. One that is more contemporary is PyGAD, a Python 3 open-source package that can be integrated with Keras and PyTorch to establish GA and optimize machine learning models. Users can rapidly construct a representation, genetic operations, and a fitness function to address a problem through GA with the aid of these packages.

2.2 Literature Review

2.2.1 Transformer as Vision Backbone

Intrigued by the breathtaking byproduct produced from transformer models in the natural language processing (NLP) community, it is adapted for vision and multimodal learning tasks. Transformer models and their variations have thus been effectively applied to a variety of tasks, e.g., image recognition [9, 50], image segmentation [51], image super-resolution [52], image generation [10], object detection [53], text-to-image synthesis [54], video understanding [55], visual question answering [56], etc.

The first study to demonstrate how transformers can completely displace traditional convolutions in deep neural networks on large-scale image datasets is Vision Transformer (ViT) [9]. It registered the initial transformer model [8] to a series of image patches that had been flattened as vectors. It was pre-trained using a huge dataset and then adjusted for recognition standards such as ImageNet classification. The model has drawn significant interest, and many new methods that expand on ViT have subsequently been developed.

Following high-level vision tasks, many transformer-based approaches have been developed for low-level vision tasks like image SR, deraining, denoising, and colorization. The image processing transformer (IPT) [10] introduced by Chen et al. is a pre-trained model built with the principle of transformer. IPT's general framework includes a joint encoder-decoder transformer core as well as multi-heads and multi-tails that can independently handle distinct tasks. Yang et al. [52] presented a transformer network for super-resolution (TTSR). It utilizes LR-HR image pairings along with reference images with material resembling that of LR images to train the model. More effective transformer-based techniques have been introduced due to the high computational complexity of the original self-attention mechanism. One of those is SwinIR [11] (our reference model) which is based on the swin transformer [12] that employs the shifted window approach.
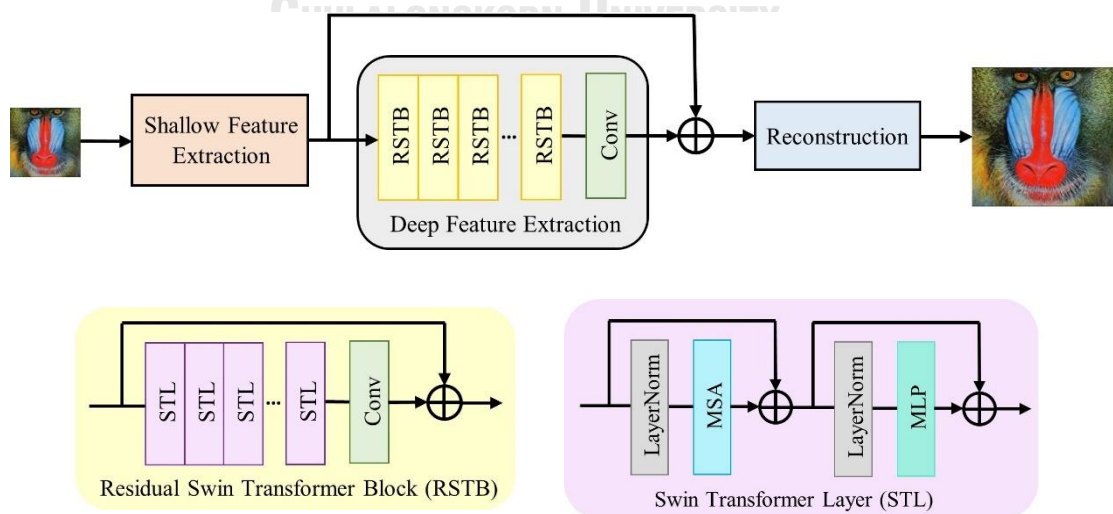
Figure 15: Overall architecture of SwinIR

2.2.2 Genetic Algorithm in Deep Learning

2.2.2.1 Genetic Algorithm for Hyperparameters Optimization

GA has been applied extensively in search problems and optimization problems [57, 58]. It is employed to optimize the parameters of the deep learning model. GA has already been used in a number of earlier studies to discover the weights [59] or the structure [60] of artificial neural networks (ANN).

In [61], Xie et al. optimize the architectures of CNN with the use of GA. The main concept behind their study is that encoding network state to a fixed-length binary string. Populations are then formed in accordance with the binary string. Additionally, every individual receives training on a reference dataset. After that, the selection stage would then be carried out after evaluating all of them. Researchers execute the GA on the CIFAR-10 dataset and observe that the yielding structures perform reasonably well. These structures have the capability to be used for image recognition tasks on a greater scale than CIFAR-10, such as the ILSVRC2012 dataset.

Cartesian genetic programming encoding is applied by Suganuma et al. [62] to optimize CNN structures for vision classification automatically. They develop tensor concatenation modules and convolutional blocks into a node function for Cartesian genetic programming. The objective of Cartesian genetic programming is recognition accuracy. The connectivity between the CNN architecture and Cartesian genetic programming is optimized. In their research, CNN architectures are developed using CIFAR-10 as the reference dataset to validate the approach. Their method is demonstrated to be capable of constructing a CNN model that is comparable to state-of-the-art models through validation.

Baldominos et al. [63] adopted GA and grammatical evolution to decrease the manual process of trial and error when defining the parameters of ConvNet (GAConvNet and GEConvNet). The ConvNet architecture and hyperparameters are determined through the evolutionary algorithm. Both the GAConvNet and GEConvNet are assessed on the benchmark dataset. According to the results, the GAConvNet and GEConvNet improve the efficiency of the conventional ConvNet.

Han et al. [64] came up with another study of applying GA to address the hyperparameter optimization issue. The fitness function in [64] combines the verification time together with the validation accuracy. The model is compacted to a single convolution layer and a single fully connected layer. The authors examine the performance of their approach with two datasets, the MNIST dataset, and the motor fault diagnosis dataset. They demonstrate how the approach can take both accuracy and the efficiency into account.

Another work introduced by Young et al. [65] is a GA-based method to select networks on multi-node clusters. They conduct the experiment of the GA to optimize the hyperparameter of a 3-layer CNN. The process of finding for hyperparameters can be substantially accelerated by the distributed GA. Real et al. [66] developed an evolutionary algorithm that only considers mutation. The deep learning model evolves over time to identify a suitable set of combinations. Due to the nature of mutation alone, the evolutionary process moves slowly. In order to optimize the hyperparameters in CNNs, Xiao et al. present a variable length GA [67]. They do not restrict the depth of the model in their work. According to experimental outcomes, they can efficiently develop hyperparameter combinations that are desirable.

2.2.2.2 Genetic Algorithm in Super-resolution

The method should make use of all relevant prior knowledge in order to overcome the super-resolution problem as efficiently as possible. For instance, balancing a flattening condition and a data quality condition was difficult. The high-frequency content will be restricted if a smoothing condition is too severe. One of the earliest studies of GA in SR is presented by Ahrens [68]. GA is implemented in two phases. The best set of registration parameters is first discovered using a GA. The point spread function parameter and the super-resolved image are then generated by another GA utilizing the registration parameters. A new SISR approach that relies on a GA and regularization prior model was proposed by Li et al. [69]. A regularization prior model that includes the non-local means (NLMs) filter, total variation (TV) and the adaptive sparse domain selection (ASDS) method for sparse representation is used to create the optimization problem. The authors integrated the GA with the iterative shrinkage

approach to handle the regularization prior model and prevent local optimization. It outperforms a number of previous state-of-the-art algorithms in terms of both numerical analysis and aesthetic effect. In [70], Kawulok et al. introduce GA-SRR, a framework with the capacity to employ a GA to tune the super-resolution reconstruction (SRR) hyperparameters. A well-known SRR algorithm [71] was optimized using average SSIM as the fitness function implemented in C++. The outcome revealed that the reconstruction process' hyperparameters are sensitive, which proves that figuring out their optimal solutions is not an easy operation.

CHAPTER III

PROPOSED METHOD

3.1 Overview of Proposed Method

SwinIR showed improvement over earlier studies, notwithstanding, the hyperparameter layout can be optimized to obtain a more suitable solution. An optimization algorithm for self-attention based SISR leveraging GA is presented in this section. Figure 16 illustrates the flowchart of the genetic algorithm-based deep multi-route self-attention for single image super-resolution (GA-MRSA).
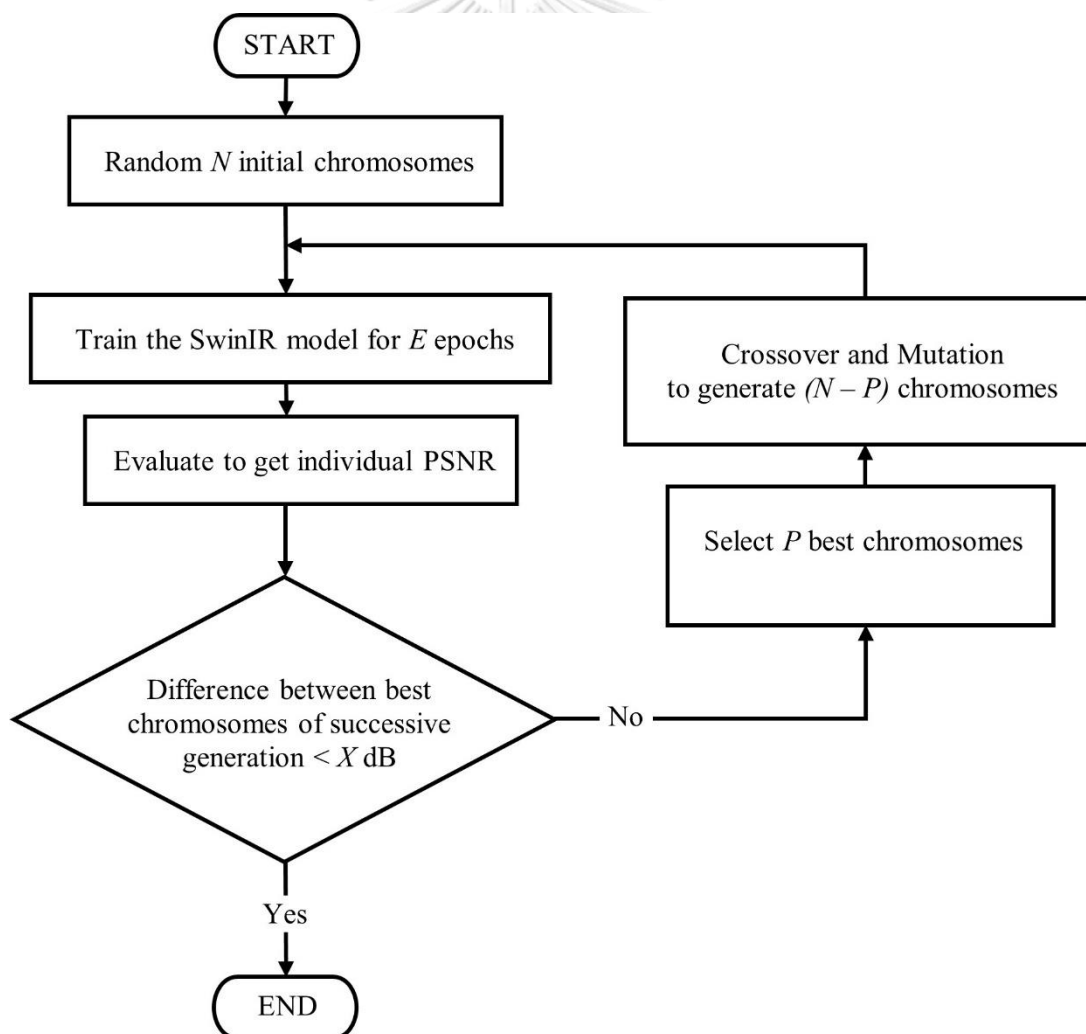
Figure 16: Flowchart of the Proposed Algorithm

3.2 Chromosome Design

The collection of genes that make up our hypothetical chromosome are hyperparameters that the SwinIR model will utilize as input to assess each individual's fitness value. The gene in this work is represented by a completely arbitrary decimal whole number. D is the problem dimensions size or the number of hyperparameters that the algorithm requires to reach the target result. A chromosome with multiple genes is derived as:

$$Chromosome = \{Gene_1, Gene_2, \ldots, Gene_D\} \tag{3.1}$$

Every chromosome represents a capable solution. In order to form the first population, N chromosomes are initialized.

$$Population = \{Chromosome_1, \ldots, Chromosome_N\}, \tag{3.2}$$

SwinIR [11], an image restoration model based on the swin transformer [12], serves as our reference baseline. Three modules build up the model architecture which comprises the shallow feature extraction, the deep feature extraction, and the upsampling modules. The deep feature extraction module is constructed of multiple residual swin transformer blocks (RSTB). SwinIR has shown outstanding promise since it integrates the benefits of CNN and Transformer. It exploits the shifted window approach to enable long-range relationships while constraining the self-attention mechanism to operate on non-superimpose regional windows.

According to the ablation experiment of the referenced paper, it can be noticed that the key parameters which are the essential determinants to impact the performance are the number of channels and the RSTB number. Consequently, we decide to tune three hyperparameters by utilizing GA. First, is the number of filters, $Gene_{f_1}$, that is used in the upsampling module. Second, $Gene_l$, the RSTB number or the number of layers in the deep feature extraction module. Last, the number of channels, $Gene_{f_2}$, which appeared in all three modules across the entire model. Therefore, our proposed chromosome can be defined as:

$$Chromosome = \{Gene_{f_1}, Gene_l, Gene_{f_2}\} \tag{3.3}$$

3.3 Fitness Function

A fitness function should be specifically described if it is expected to address any issue. After each repetition, the fitness function produces a single quantitative number that represents the utility of each chromosome. The fitness function for this proposed SISR optimization approach is based upon the primary objective function, peak signal-to-noise ratio (PSNR) of the RGB channels.

By providing LR image I_{LR} as an input, $H_{SFE}(\cdot)$ is the convolution layer function applied to extract low-frequency information F_{SF}.

$$F_{SF} = H_{SFE}(I_{LR}) \tag{3.4}$$

Next, the deep feature extraction function $H_{DFE}(\cdot)$ is used to preserve higher frequency details F_{DF}.

$$F_{DF} = H_{DFE}(F_{SF}) \tag{3.5}$$

After the summation of the shallow feature F_{SF} and the deep feature F_{DF}, we generate the reconstructed super-resolved image I_{SR} in (3.6).

$$I_{SR} = H_{UP}(F_{SF} + F_{DF}), \tag{3.6}$$

at which the upsampling module function is denoted by $H_{UP}(\cdot)$. The model is trained by minimizing the pixel-wise L_1 loss function during the training procedure (mentioned in section 2.1.2.3).

$$\mathcal{L}_{L_1} = \|I_{SR} - I_{GT}\|, \tag{3.7}$$

where I_{GT} is the original HR image before downsampled, or in other words, the ground truth image.

PSNR [36] is among the most popular reconstruction performance evaluation. The unit is in decibels or dB. The mean squared error (MSE) across images is a key component in the definition of PSNR for SISR. The MSE between I and $\hat{I}$ can be calculated with Equation (3.8), given that $\hat{I}$ stands for the reconstructed SR image and I is the HR ground truth image.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left[I(i,j) - \hat{I}(i,j) \right]^2, \tag{3.8}$$

Where n and m represent the height and width of the image, respectively. Provided L as the maximum pixel value, usually employed 255 in the incident of using the 8-bit description. The PSNR value is then denoted as follows:

$$PSNR = 10 \times \log_{10} \left(\frac{L^2}{MSE} \right) \tag{3.9}$$

PSNR typically underperforms when striving to portray the reconstruction efficiency in a real-world scenario because it only focuses on differences between correlated pixels and not the visual experience. Although there are currently no perfectly accurate perceptual metrics, PSNR persists to be the most commonly used assessment criterion for SR methods since it is vital for comparison to other scientific research.

3.4 Selection, Crossover, and Mutation

Selection: Subsequent to evaluating the solution to attain the fitness value, PSNR will be arranged from the best to the worst. The higher PSNR value indicates better performance. As a result, the best P chromosomes are chosen as parents to construct the new children.

$$Parents = \{Parent_1, Parent_2, \dots, Parent_P\} \tag{3.10}$$

$$Best_G = Parent_1$$

Better solutions are retained, whereas inferior ones are eliminated under the principle of natural selection.

Crossover: In a process known as a crossover, more than one parents unite their genetic characteristics to produce a new offspring possessing their traits. Strong chromosome crossover may not always lead to a better outcome, but the likelihood of it being great is large. To create $(N - P)$ new offspring, we crossover all genes of the parents.

$$Crossover\ Index = rand(0, P - 1) \tag{3.11}$$

$$C = \begin{cases} Parent_1, if\ Crossover\ Index = 0 \\ Parent_2, if\ Crossover\ Index = 1 \\ Parent_P, if\ Crossover\ Index = P - 1 \end{cases}$$

$$Children = \{Child_1, Child_2, \dots, Child_{(N-P)}\}$$

$$Child = \{C_1, C_2, \dots, C_D\}$$

Mutation: To sustain the diversity of the population in a new generation, the genes in the offspring are purposefully altered by random solution space with a mutation index. A new population with the same number of chromosomes is generated by merging parents and the offspring achieved after the mutation process.

$$Mutation\ Index = rand(0, D - 1) \tag{3.12}$$

$$M = \begin{cases} Change\ 1\ gene, if\ Mutation\ Index = 0 \\ Change\ 2\ genes, if\ Mutation\ Index = 1 \\ Change\ D\ genes, if\ Mutation\ Index = D - 1 \end{cases}$$

3.5 Termination Criteria

In GA, a variety of termination criteria are commonly utilized. We propose a procedure in which the GA will end once the fitness value of the best chromosome of the current generation is higher than the fitness value of the best chromosome of the previous generation by less than X, it will halt.

$$Stop\ if\ (|Best_G - Best_{G-1}| < X) \tag{3.13}$$

3.6 Genetic Algorithm-Based Deep Multi-Route Self-Attention Network

In correspondence to Figure 16, the algorithm starts by initializing the population of the first generation. The population of generation G is derived in Equation (3.14).

$$Population_G = \{Chromosome_1, \dots, Chromosome_N\} \qquad (3.14)$$

Subsequently, each random chromosome should be injected into the fitness function to assess the performance separately. SwinIR will be trained using E epochs, and its fitness will be evaluated using the PSNR evaluation metric. This could be expressed as

$$Fitness_i = PSNR\left(H_{SwinIR_i}(I_{LR})\right). \qquad (3.15)$$

Given that the SwinIR and PSNR function of the i-th chromosome ($i = 1, 2, \dots, N$) is denoted by $H_{SwinIR_i}(\cdot)$ and $PSNR_i(\cdot)$, respectively. The maximum PSNR of that specific generation was then acquired, and the termination criteria were subsequently inspected. If the highest PSNR fails the requirement, the loop will keep proceeding. The successive phase is selection, where each chromosome is sorted depending on its PSNR value, and the best P chromosomes are chosen as the parents. Provided that the current generation is indicated by G, it can be denoted in Equation (3.16).

$$Parents_G = Selection(Population_G, Fitness_G), \qquad (3.16)$$

After employing crossover to the parents and accompanied by mutation, $(N - P)$ offspring are produced.

$$Children_G = Mutation(Crossover(Parents_G)) \qquad (3.17)$$

The new population of size N is acquired by concatenating the current P parents and $(N - P)$ children.

$$Population_{G+1} = Parents_G + Children_G \qquad (3.18)$$

Eventually, the novel population is taken as the input to evaluate the fitness value of the subsequent generation. The repetitive procedure would continue incessantly unless the termination condition were met. As an outcome, we were able to obtain a set of hyperparameters that achieved our objective aim.

CHAPTER IV
EXPERIMENTS AND RESULTS

4.1 Experimental Setup

4.1.1 Genetic Algorithm

All three of the hyperparameters of the chromosome are constructed with constraints. It is important to narrow down the search scope with the use of all available information. Thus, it benefits in avoiding populations from becoming trapped at local minima with incorrect features and quickens the investigation process. $Gene_{f_1}$ and $Gene_{f_2}$ are the numbers of filters in the upsampling and deep feature extraction module, where random searches are restricted from 32 to 128 and 60 to 240, respectively. The original model's RSTB number $Gene_l$ was determined as 6. We set the search from 3 to 6 layers. The assigned values for the number of chromosomes N and the number of parents P in a population are 30 and 5, in the particular order. Therefore, every generation produces 25 new children.

4.1.2 Single Image Super-Resolution

4.1.2.1 SISR Settings

In the phase of the fitness function, 30 chromosomes must be passed through the SwinIR model one by one to train for E epochs. Following that, assess individual performance with PSNR. This step is the principal reason that causes the proposed method to consume a considerably high computational time. Taking the time required and whether the outcomes are comparable into account, we assign the number of epochs E to 100. Consequently, we also utilize the training seed. The way computers random a value distinct from a typical random method; they can always produce the same value using a seed. Every time we retrain a model, the model's initial parameters—which are random—change. Determining the seed help operating the trained result always to complete the same. In a longer duration of the training, it does not affect much. Nevertheless, a fixed seed is required for a fair comparison because we only train each model for 100 epochs.

The number of swin transformer layers (STL), attention head number, and window size for SwinIR is set to 6, 6, and 8, respectively. 8 LR image patches of 48 × 48 are inputs for each training mini-batch. Train by using the ADAM optimizer [72] with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The initial learning rate is set to 2e-4. The proposed method is implemented with Python 3.9 as the programming language and PyTorch v1.11.0 library on an NVIDIA GeForce RTX 3090 Ti.

4.1.2.2 Datasets

❖ **Training**

Following [17, 27, 73], the training is performed on the DIV2K [74]. DIV2K or DIVerse 2K resolution image dataset is a PNG image dataset that served as a benchmark for the NTIRE 2017 challenge. It includes 800 training, 100 validation, and 100 testing images with an average resolution (1972, 1437). Some example images from the DIV2K dataset are shown in Figure 17. For this study, we only used 800 HR images as the ground truth and bicubically downscaled them by a scale factor of 2, 3, and 4 to get the LR input images.

Figure 17: Example images from the DIV2K dataset

❖ **Testing**

First, we test on the freely accessible Set5 [75], which is a standard benchmark dataset comprising five images: baby, bird, butterfly, head, and woman.

(b) bird

(c) butterfly

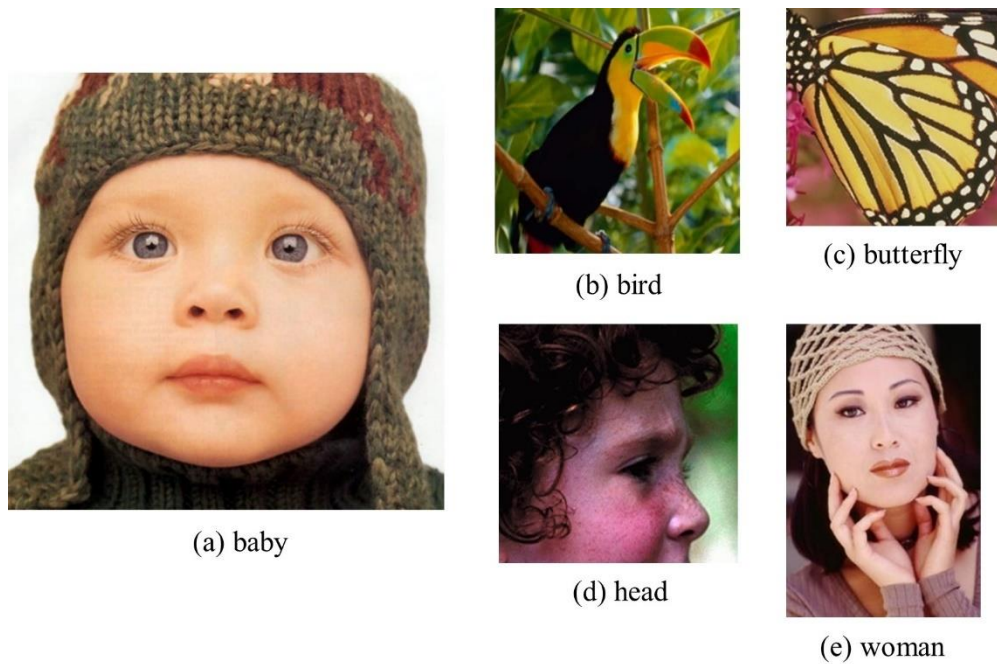(a) baby

(d) head

(e) woman

Figure 18: Images from the Set5 dataset

Additionally, the performance is also evaluated on the Set14 testing datasets [76]. Set14 consists of 14 images which cover more categories than the Set5 dataset.

Figure 19: Images from the Set14 dataset

4.1.2.3 Evaluation Metrics

The reconstructed results are evaluated using quantitative evaluation metrics, peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). Both PSNR and SSIM for the testing are on the luminance (Y) channel of the image. PSNR is mentioned in section 3.3 and derived as in Equations (3.8) and (3.9).

Take into consideration that our human visual system is greatly suitable for drawing out image's structure information. The structural similarity index measure (SSIM) [36] is introduced for evaluating the structural similarity between images based on three distinct comparisons, i.e., in terms of brightness, contrast, and structures. x and y are images with N pixels, SSIM can be computed as:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{4.1}$$

Where the brightness μ_x, μ_y and contrast σ_x, σ_y are approximated as the mean average and the standard deviation of x, y, respectively. In addition, σ_{xy} is the covariance between x and y, c_1 and c_2 are terms of constant. SSIM values are in the range of 0 to 1. Closer to 1 indicates that the reconstructed image has a higher similarity to the ground truth image.

4.2 Experimental Results

In order to maximize the validation PSNR, GA is applied to the SwinIR. As previously stated, when the PSNR of the best chromosome in that current generation is higher than the best chromosome of the previous generation by less than 0.02 dB, the recursive process will come to an end. The three chromosome characteristics ($Gene_{f1}$, $Gene_l$, and $Gene_{f2}$) and the RGB PSNR of the SwinIR and the model from the proposed method (GA-MRSA) at a scale factor of 2 after training for 100 epochs on Set5 are shown in Table 1.

Table 1: Chromosome Characteristics and PSNR of SwinIR and the Proposed
Method on Set5 at Scale Factor 2

Model	Chromosome			Set5 images	PSNR	Avg PSNR
	$Gene_{f_1}$	$Gene_l$	$Gene_{f_2}$			
SwinIR [11]	64	6	180	baby	36.73	34.78
				bird	38.65	
				butterfly	32.12	
				head	32.03	
				woman	34.36	
GA-MRSA	110	6	168	baby	36.99	34.99
				bird	39.04	
				butterfly	32.43	
				head	32.14	
				woman	34.37	

A model with 110 filters in the upsampling module, 6 RSTB blocks, and 168 channels across the entire network is the more suitable outcome from the proposed GA approach. The average RGB channel PSNR of the default SwinIR at a scale factor of 2 after training for 100 epochs and testing on the Set5 dataset is 34.78 dB. A 34.99 dB RGB channel PSNR on the Set5 dataset was attained from the comparable proposed GA-MRSA. It can be seen that the PSNR of the proposed method outperformed that of the reference model by 0.21 dB, where all five images in the Set5 dataset achieved higher.

Table 2 depicts a quantitative comparison of PSNR and SSIM values (both on the luminance channel) between the reference model and the solution of the proposed optimization technique at a scale factor of 2 on the Set5 and Set14 testing datasets. After 400 epochs of training, the average PSNR of the reference SwinIR at a scale factor of 2 on the Set5 testing dataset is 37.60 dB. We can notice that the average PSNR on Set5 and Set14 increases by 0.14 and 0.09 dB in respectful order.

Table 2: Quantitative Comparison of SwinIR and Proposed Method at Scale Factor 2

Model	Set5		Set14	
	PSNR_Y	SSIM_Y	PSNR_Y	SSIM_Y
SwinIR [11]	37.60	0.9598	33.29	0.9166
GA-MRSA	37.74	0.9596	33.38	0.9167

The visual comparison of the original HR, after upsampling with bicubic interpolation, SwinIR, and the proposed method solution at a scale factor of 2 performing on a crop section of Set5's butterfly and Set14's coastguard is presented in Figure 20 and Figure 21, respectively.

(a) HR   (b) Bicubic

(c) SwinIR   (d) GA-MRSA

Figure 20: Visual Comparison of SwinIR and Proposed Method at Scale Factor 2 on Set5's butterfly
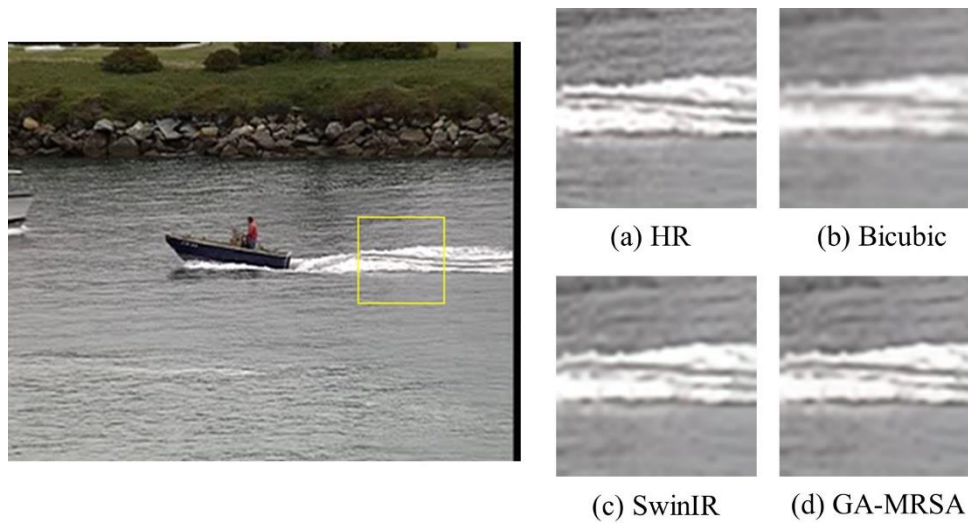
Figure 21: Visual Comparison of SwinIR and Proposed Method at Scale Factor 2 on Set14's coastguard

In Figure 21, the model from the proposed method can better preserve the wave ridge on the left side of the coastguard crop patch from the ground truth image compared to the SwinIR.

The quantitative comparison of SwinIR and the model from the proposed method at a scale factor of 3 on Set5 and Set14 is displayed in Table 3. The performance increases by 0.04 dB and 0.02 dB in terms of PSNR on Set5 and Set14, respectively.

Table 3: Quantitative Comparison of SwinIR and Proposed Method at Scale Factor 3

Model	Set5		Set14	
	PSNR_Y	SSIM_Y	PSNR_Y	SSIM_Y
SwinIR [11]	33.90	0.9240	30.10	0.8396
GA-MRSA	33.94	0.9240	30.12	0.8396

The qualitative comparisons of the HR, bicubic interpolation, SwinIR, and model from the proposed method at the scale factor of 3 on Set5's baby, Set5's bird, and Set14's zebra are presented in Figure 22, Figure 23, and Figure 24, respectively. We can notice that the model from our proposed method reconstructed finer edges and details
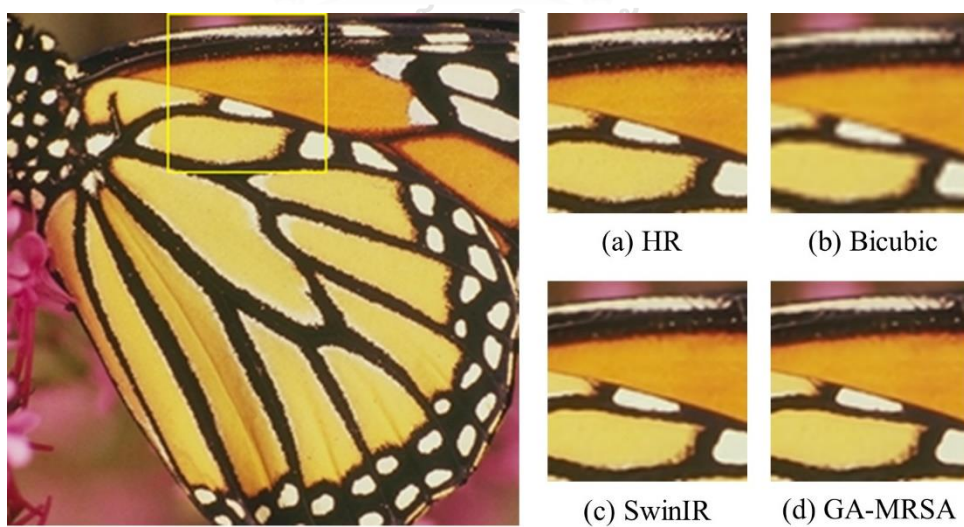

Figure 22: Visual Comparison of SwinIR and Proposed Method at Scale Factor 3 on Set5's baby

Figure 23: Visual Comparison of SwinIR and Proposed Method at Scale Factor 3 on Set5's bird

(a) HR   (b) Bicubic

(c) SwinIR   (d) GA-MRSA

Figure 24: Visual Comparison of SwinIR and Proposed Method at Scale Factor 3 on Set14's zebra

From the visual comparison, it can be pointed out that a high number of upsampling filters leads to an insightful view of various varieties of changes in the scene for a better human visual system.

Table 4 shows the numerical comparison of SwinIR and the model from the proposed method at a scale factor of 4 on Set5 and Set14 testing datasets. The model from our proposed method outperformed the SwinIR by 0.02 dB and 0.07 dB in terms of PSNR on Set5 and Set14, in the proper order.

Table 4: Quantitative Comparison of SwinIR and Proposed Method at Scale Factor 4

Model	Set5		Set14	
	PSNR_Y	SSIM_Y	PSNR_Y	SSIM_Y
SwinIR [11]	31.63	0.8878	28.25	0.7755
GA-MRSA	31.65	0.8877	28.32	0.7762

It is obviously noticeable that the GA-MRSA can generate more correct information when compared to the SwinIR. GA-MRSA can reconstruct clearer and sharper edges of the text "How" in the Set14's ppt3 than the SwinIR.
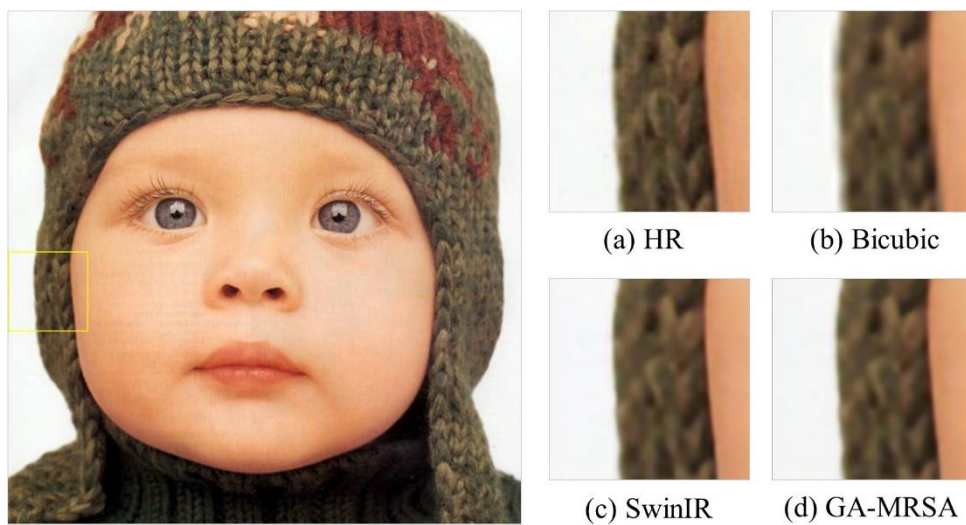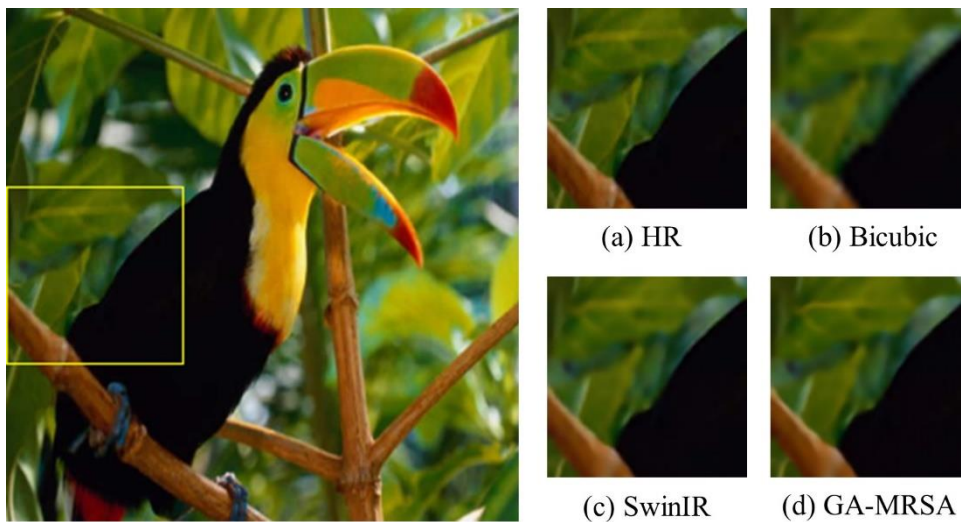

Figure 25: Visual Comparison of SwinIR and Proposed Method at Scale Factor 4 on Set5's bird

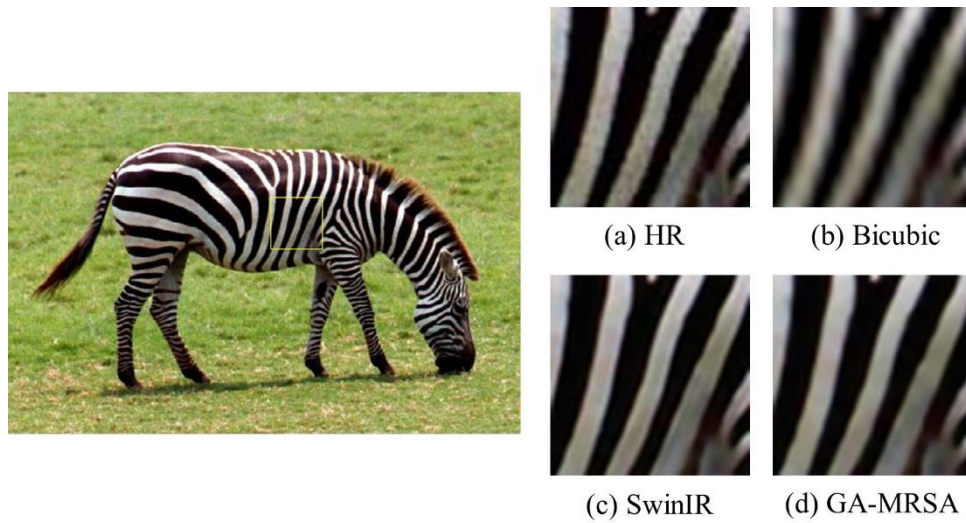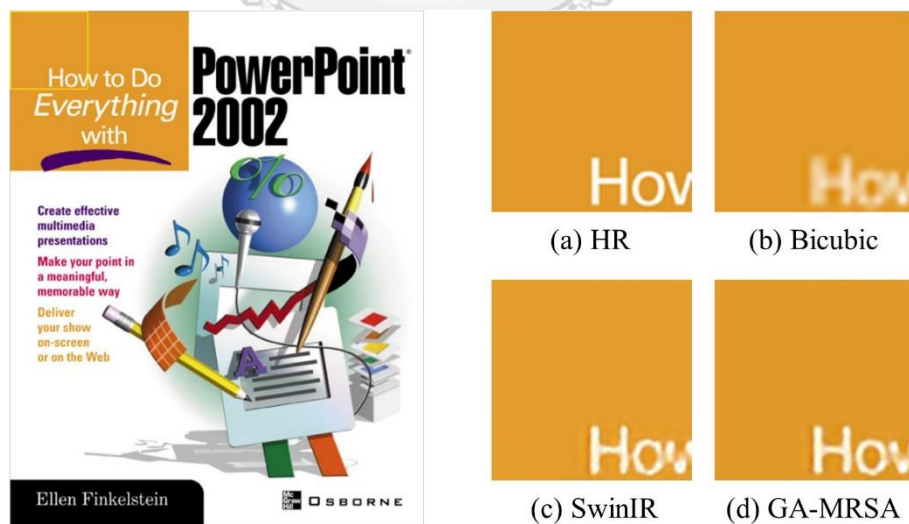Figure 26: Visual Comparison of SwinIR and Proposed Method at Scale Factor 4 on Set14's ppt3

Figure 27 and Figure 28 demonstrate the visual comparison of the HR, after bicubic interpolation, SwinIR, and GA-MRSA performed on Set14's zebra crop patches of size 60×60 and 100×100, respectively. In Figure 27, GA-MRSA reconstructed the white strips on the zebra's leg in the correct direction, while SwinIR reconstructed them in the wrong direction.

(a) HR   (b) Bicubic

(c) SwinIR   (d) GA-MRSA
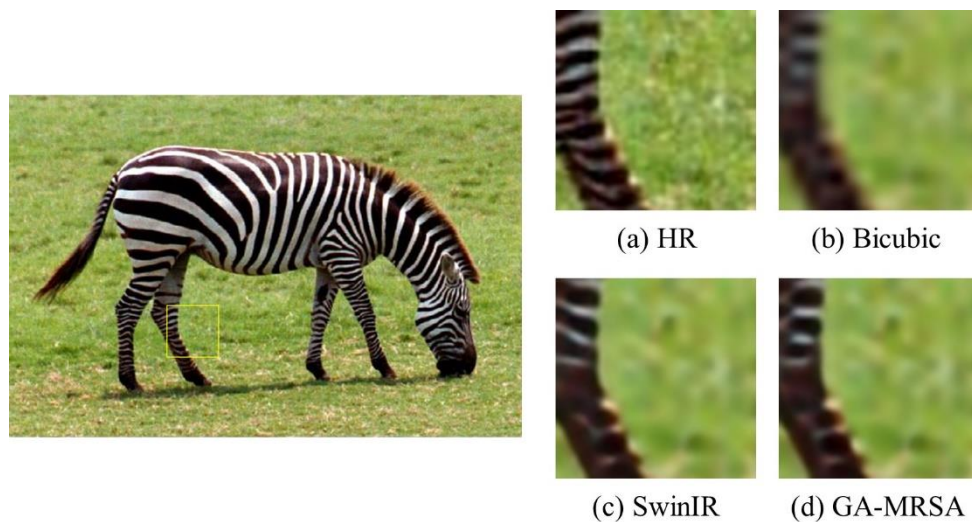
Figure 27: Visual Comparison of SwinIR and Proposed Method at Scale Factor 4 on Set14's zebra (60×60)

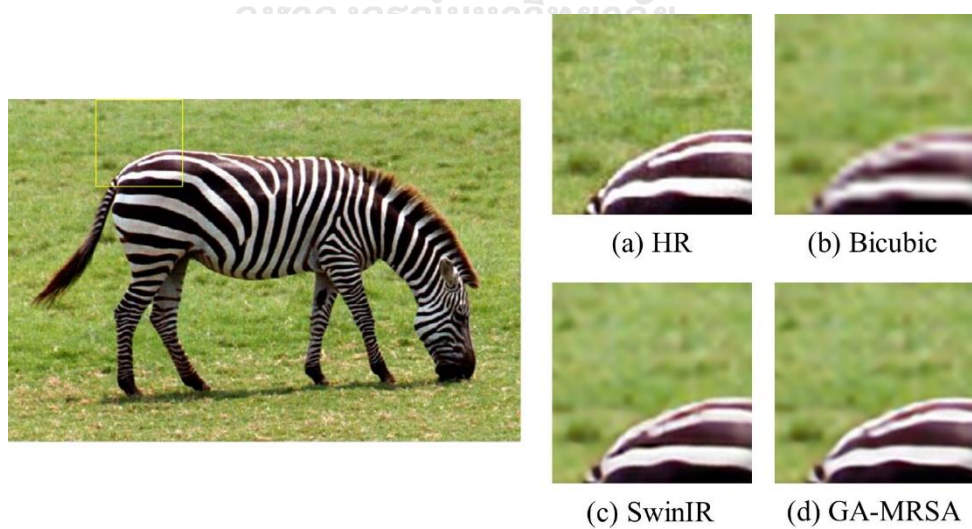(a) HR   (b) Bicubic

(c) SwinIR   (d) GA-MRSA

Figure 28: Visual Comparison of SwinIR and Proposed Method at Scale Factor 4 on Set14's zebra (100×100)

4.3 Results Discussion

We can notice that the number of parameters of the model from the proposed method also decreases from the number of parameters of the baseline model. Table 5 shows the comparison of the overall number of parameters between SwinIR and our proposed method at a scale factor of 2. It is reduced from 11.7 million parameters to 10.6 million parameters, which is around 1.1 million parameters or approximately 9.5 %. This proves to us that by modifying these three hyperparameters, it is possible to achieve higher performance with fewer number of parameters.

Table 5: Comparison of Overall Parameters between SwinIR and Proposed Method at Scale Factor 2

Model	Number of Parameters
SwinIR [11]	11,752,487
GA-MRSA	10,633,747

From the experimental setup, the upper boundary of the model is constrained to limit the size of the random search space. The chromosome characteristics of the Upper bound are 128, 6, and 240 for $Gene_{f_1}$, $Gene_l$, and $Gene_{f_2}$, respectively. The models are shown with their corresponding chromosome characteristics and the number of parameters in Table 6. The Upper bound has around 10 million more parameters than the SwinIR, which more parameters are also indicating more computational cost.

Table 6: Chromosome Characteristics and Total Parameters of SwinIR and Upper bound at Scale Factor 3

Model	Chromosome			Number of Parameters
	$Gene_{f_1}$	$Gene_l$	$Gene_{f_2}$	
SwinIR [11]	64	6	180	11,937,127
Upper bound	128	6	240	21,978,923

The quantitative comparison of SwinIR and the Upper bound that is mentioned above at a scale factor of 3 in terms of PSNR and SSIM on Set5, and Set14 testing datasets are shown in Table 7. Regarding PSNR, the Upper bound is less than SwinIR for 1.32 dB and 0.97 dB on Set5 and Set14. While also worse than SwinIR for 0.0164 and 0.0197 in terms of SSIM. This shows us that a model with a higher number of parameters does not guarantee better performance. The graph of PSNR comparison between SwinIR and the Upper bound at a scale factor of 3 on Set5 is illustrated in Figure 29.

Table 7: Quantitative Comparison of SwinIR and Upper bound at Scale Factor 3

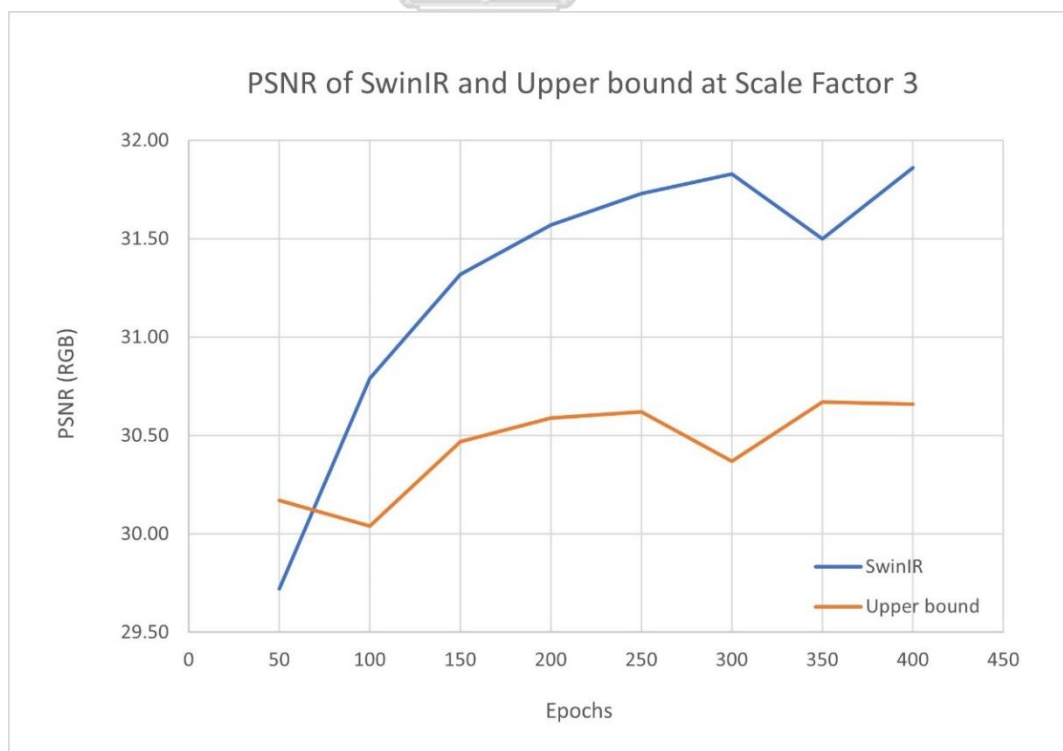| Model	Set5		Set14	
	PSNR_Y	SSIM_Y	PSNR_Y	SSIM_Y
SwinIR [11]	33.90	0.9240	30.10	0.8396
Upper bound	32.58	0.9076	29.18	0.8199

Figure 29: PSNR of SwinIR and Upper bound at Scale Factor 3 for 400 Epochs

CHAPTER V
CONCLUSION AND FUTURE WORK

5.1 Conclusion

The proposed GA-MRSA is a genetic algorithm-based transformer-based optimization method for single-image super-resolution that aims to focus on three core components in SwinIR to generate reconstructed high-resolution images with the higher visual quality compared to the baseline method. Besides, the stochastic hyperparameter approach is applied to form the population of GA in this work within the restricted search space. The proposed framework performs hyperparameter optimization while minimizing the loss function to obtain a feasible network structure. Experimental results clearly indicate that our GA-MRSA achieves a higher PSNR of up to 0.14 dB and an average of 0.06 dB in the test sets compared to the state-of-the-art. Additionally, the proposed method can produce insightful views of various changes in the scene and smooth output images for the human visual system. Moreover, the proposed GA-MRSA provides lower computational complexity and number of parameters than SwinIR.

5.2 Future Work

According to current performance, the proposed method can be further enhanced in the future to integrate the inception hyperparameter optimization in the network structure design of SISR models. This design can obtain high-and-low-level feature sharing in successive layers to construct a good reconstructed image. Furthermore, the computational complexity should be considered by assembling the evolutionary algorithm with the multi-model above.

REFERENCES

[1]   H. Greenspan, "Super-resolution in medical imaging," *The computer journal,* vol. 52, no. 1, pp. 43-63, 2009.

[2]   Y. Chen, Y. Tai, X. Liu, C. Shen, and J. Yang, "Fsrnet: End-to-end learning face super-resolution with facial priors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2492-2501.

[3]   L. Zhang, H. Zhang, H. Shen, and P. Li, "A super-resolution reconstruction algorithm for surveillance images," *Signal Processing,* vol. 90, no. 3, pp. 848-859, 2010.

[4]   H. Zhang, Z. Yang, L. Zhang, and H. Shen, "Super-resolution reconstruction for multi-angle remote sensing images considering resolution differences," *Remote Sensing,* vol. 6, no. 1, pp. 637-657, 2014.

[5]   J. Caballero *et al.*, "Real-time video super-resolution with spatio-temporal networks and motion compensation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4778-4787.

[6]   R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE transactions on pattern analysis and machine intelligence,* vol. 38, no. 1, pp. 142-158, 2015.

[7]   A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM,* vol. 60, no. 6, pp. 84-90, 2017.

[8]   A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems,* vol. 30, 2017.

[9]   A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929,* 2020.

[10]   H. Chen *et al.*, "Pre-trained image processing transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12299-12310.

[11]   J. Liang *et al.*, "Swinir: Image restoration using swin transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1833-1844.

[12]   Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10012-10022.

[13]   C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European conference on computer vision*, 2014: Springer, pp. 184-199.

[14]   C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *European conference on computer vision*, 2016: Springer, pp. 391-407.

[15]   W. Shi *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1874-1883.

[16]   C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681-4690.

[17] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 136-144.

[18] T. Tong, G. Li, X. Liu, and Q. Gao, "Image super-resolution using dense skip connections," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4799-4807.

[19] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep laplacian pyramid networks for fast and accurate super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 624-632.

[20] M. Haris, G. Shakhnarovich, and N. Ukita, "Deep back-projection networks for super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1664-1673.

[21] Z. Li *et al.*, "Feedback network for image super-resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3867-3876.

[22] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence,* vol. 38, no. 2, pp. 295-307, 2015.

[23] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646-1654.

[24] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4539-4547.

[25] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3147-3155.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[27] Y. Zhang *et al.*, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 286-301.

[28] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," *Advances in neural information processing systems,* vol. 29, 2016.

[29] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1637-1645.

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700-4708.

[31] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132-7141.

[32] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang, "Second-order attention network for single image super-resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11065-11074.

[33] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," *arXiv preprint arXiv:1903.10082,* 2019.

[34] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*, 2016: Springer, pp. 694-711.

[35] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on computational imaging,* vol. 3, no. 1, pp. 47-57, 2016.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing,* vol. 13, no. 4, pp. 600-612, 2004.

[37] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414-2423.

[38] I. Goodfellow *et al.*, "Generative adversarial networks," *Communications of the ACM,* vol. 63, no. 11, pp. 139-144, 2020.

[39] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805,* 2018.

[40] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[41] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.,* vol. 21, no. 140, pp. 1-67, 2020.

[42] T. Back, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[43] D. E. Golberg, "Genetic algorithms in search, optimization, and machine learning," *Addion wesley,* vol. 1989, no. 102, p. 36, 1989.

[44] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, 1995, vol. 4: IEEE, pp. 1942-1948.

[45] K. V. Price, "Differential evolution," in *Handbook of optimization*: Springer, 2013, pp. 187-214.

[46] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *science,* vol. 220, no. 4598, pp. 671-680, 1983.

[47] L. M. Schmitt, "Theory of genetic algorithms," *Theoretical Computer Science,* vol. 259, no. 1-2, pp. 1-61, 2001.

[48] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[49] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[50] H. Touvron *et al.*, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning*, 2021: PMLR, pp. 10347-10357.

[51]   L. Ye, M. Rochan, Z. Liu, and Y. Wang, "Cross-modal self-attention network for referring image segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 10502-10511.

[52]   F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, "Learning texture transformer network for image super-resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5791-5800.

[53]   N. Carion *et al.*, "End-to-end object detection with transformers," in *European conference on computer vision*, 2020: Springer, pp. 213-229.

[54]   A. Ramesh *et al.*, "Zero-shot text-to-image generation," in *International Conference on Machine Learning*, 2021: PMLR, pp. 8821-8831.

[55]   C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "Videobert: A joint model for video and language representation learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7464-7473.

[56]   H. Tan and M. Bansal, "Lxmert: Learning cross-modality encoder representations from transformers," *arXiv preprint arXiv:1908.07490,* 2019.

[57]   C. R. Reeves, "A genetic algorithm for flowshop sequencing," *Computers & operations research,* vol. 22, no. 1, pp. 5-13, 1995.

[58]   C. R. Houck, J. Joines, and M. G. Kay, "A genetic algorithm for function optimization: a Matlab implementation," *Ncsu-ie tr,* vol. 95, no. 09, pp. 1-10, 1995.

[59]   X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE,* vol. 87, no. 9, pp. 1423-1447, 1999.

[60]   K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary computation,* vol. 10, no. 2, pp. 99-127, 2002.

[61]   L. Xie and A. Yuille, "Genetic cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1379-1388.

[62]   M. Suganuma, S. Shirakawa, and T. Nagao, "A genetic programming approach to designing convolutional neural network architectures," in *Proceedings of the genetic and evolutionary computation conference*, 2017, pp. 497-504.

[63]   A. Baldominos, Y. Saez, and P. Isasi, "Evolutionary convolutional neural networks: An application to handwriting recognition," *Neurocomputing,* vol. 283, pp. 38-52, 2018.

[64]   J.-H. Han, D.-J. Choi, S.-U. Park, and S.-K. Hong, "Hyperparameter optimization using a genetic algorithm considering verification time in a convolutional neural network," *Journal of Electrical Engineering & Technology,* vol. 15, no. 2, pp. 721-726, 2020.

[65]   S. R. Young, D. C. Rose, T. P. Karnowski, S.-H. Lim, and R. M. Patton, "Optimizing deep learning hyper-parameters through an evolutionary algorithm," in *Proceedings of the workshop on machine learning in high-performance computing environments*, 2015, pp. 1-5.

[66]   E. Real *et al.*, "Large-scale evolution of image classifiers," in *International Conference on Machine Learning*, 2017: PMLR, pp. 2902-2911.

[67]   X. Xiao, M. Yan, S. Basodi, C. Ji, and Y. Pan, "Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm," *arXiv preprint arXiv:2006.12703,* 2020.

[68]   B. Ahrens, "Genetic algorithm optimization of super-resolution parameters," in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, 2005, pp. 2083-2088.

[69]   Y. Li *et al.*, "Single image super-resolution reconstruction based on genetic algorithm and regularization prior model," *Information Sciences,* vol. 372, pp. 196-207, 2016.

[70]   M. Kawulok, D. Kostrzewa, P. Benecki, and L. Skonieczny, "Optimizing Super-resolution Reconstruction using a Genetic Algorithm," in *ICAART (2)*, 2018, pp. 599-605.

[71]   S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE transactions on image processing,* vol. 13, no. 10, pp. 1327-1344, 2004.

[72]   D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.

[73]   Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2472-2481.

[74]   E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 126-135.

[75]   M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," 2012.

[76]   R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*, 2010: Springer, pp. 711-730.

VITA

NAME   Nisawan Ngambenjavichaikul

DATE OF BIRTH   6 April 2005

PLACE OF BIRTH   Bangkok, Thaialnd

INSTITUTIONS   Bachelor of Engineering (Electrical Engineering),
ATTENDED   Sripatum University, 2017 - 2020
HOME ADDRESS   2022/15 Phaholyothin Road,
Sena Nikhom, Chatuchak,
Bangkok, Thailand 10900
PUBLICATION   N. Ngambenjavichaikul, S. Chen, and S. Aramvith, "Optimal Deep Multi-Route Self-Attention for Single Image Super-Resolution", In Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2022, pp. 1182 - 1187.
AWARD RECEIVED -