Real-time image super-resolution reconstruction for system on chip FPGA

Watchara Ruangsang

Faculty of Engineering

Follow this and additional works at: https://digital.car.chula.ac.th/chulaetd

Recommended Citation

This Thesis is brought to you for free and open access by Chula Digital Collections. It has been accepted for inclusion in Chulalongkorn University Theses and Dissertations (Chula ETD) by an authorized administrator of Chula Digital Collections. For more information, please contact ChulaDC@car.chula.ac.th.
REAL-TIME IMAGE SUPER-RESOLUTION RECONSTRUCTION FOR SYSTEM ON CHIP FPGA

Mr. Watchara Ruangsang

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Electrical Engineering
Department of Electrical Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2021
Copyright of Chulalongkorn University
การสร้างคืนภาพความละเอียดสูงยิ่งยวดเวลาจริงสำหรับแอพพิจิôtแบบฮับที่มีระบบประมวลผล

นายวัชระ เรืองสังข์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2564
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย
Thesis Title: REAL-TIME IMAGE SUPER-RESOLUTION RECONSTRUCTION FOR SYSTEM ON CHIP FPGA

By: Mr. Watchara Ruangsang

Field of Study: Electrical Engineering

Thesis Advisor: Associate Professor SUPAVADEE ARAMVITH, Ph.D.

Thesis Co Advisor: Professor Takao Onoye, D.Eng.

Accepted by the FACULTY OF ENGINEERING, Chulalongkorn University in Partial Fulfillment of the Requirement for the Doctor of Philosophy

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

DISSERTATION COMMITTEE

Chairman
(Professor Prasit Prapingmongkolkarn, Ph.D.)

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

Thesis Co-Advisor
(Professor Takao Onoye, D.Eng.)

Examiner
(Assistant Professor SUREE PUMRIN, Ph.D.)

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

External Examiner
(Sitapa Rujikietgumjorn, Ph.D.)
การสร้างคืนภาพความละเอียดสูงยิ่งยวดเวลาจริงสำหรับเอฟพีจีเอแบบชิปที่มีระบบประมวลผล (REAL-TIME IMAGE SUPER-RESOLUTION RECONSTRUCTION FOR SYSTEM ON CHIP FPGA)

ที่ปรึกษาหลัก:
รศ. ดร. สุภาวดี อร่ามวิทย์
ที่ปรึกษาร่วม:
ศ. ดร. ทาคาโอะ โอโนเย

ในปัจจุบัน การสร้างคืนภาพความละเอียดสูงยิ่งยวดใช้เทคนิคภายใต้โครงข่ายประสาทเทียมแบบคอนโวลูชั่นได้รับความสนใจอย่างมากในการใช้งานด้านคอมพิวเตอร์วิชชั่นและบริษัทที่ทำเกี่ยวกับปัญหาประดิษฐ์ อย่างไรก็ตามวิธีการการสร้างคืนภาพความละเอียดสูงยิ่งยวดต้องใช้ทรัพยากรในการประมวลที่ต้องใช้พลังงานสูง และการใช้หน่วยความจำจำนวนมาก ซึ่งปัญหาที่สำคัญในการนำไปใช้งานจริง คือ การออกแบบให้มีประสิทธิภาพและไม่แตกต่างคาดหมาย ปรับปรุงความคมชัดและคุณภาพภาพออกมาไม่เสียหาย โดยการแก้ไขดังกล่าว เราจึงเสนอโครงข่ายชั่วชีวูธ์สัญญาณตกค้างร่วมกับทางเดียว จากการใช้มิติของสัญญาณตกค้างในสัญญาณตกค้างภายใต้การเชื่อมต่อหลายทาง เฟรมที่กลับล่าช้าช่วงเวลาของขั้นตอนการส่งข้อมูลก่อนหน้าในการสร้างคืนภาพความละเอียดสูง โดยการเรียงแบบคืนในมิติของสัญญาณทำให้สามารถส่งข้อมูลจากการใช้การประมวลผลภาพ และนำไปใช้งานกับเอฟพีจีเอแบบชิปที่มีระบบประมวลผลโดยการใช้เทคนิคระบบนำข้อมูล แบบอิงชั่วชีวูธ์ที่เป็นไม่ยุ่งยากขั่นแบบจำนวนเต็ม ไปประมวลผลโดยใช้เพดิดพิธ์ โดยผลการทดลองแสดงให้เห็นว่า วิธีการที่นำเสนอสามารถลดขนาดขั้นตอนการใช้งานตัวเลขที่สูง (8.4 เท่าเมื่อเทียบกับวิธีการแบบออร์เจีย) โดยที่รักษาคุณภาพภาพออกและค่าพีเอสเอ็นอาร์กับเทคนิคทั่วไปอย่างเช่นกัน ถ้าสามารถนำไปประมวลผลที่เอฟพีจีเอแบบชิปที่มีระบบประมวลพลังด้านความเร็ว 30 เฟรมต่อวินาที และภาพมีความถูกต้องในการใส่แบบการ์ดแบบที่เอฟพีจีเอไม่ลดลงน้อยกว่า 1 เดซิเบล

สาขาวิชา วิศวกรรมไฟฟ้า ลายมือชื่อนิสิต ....................................................
ปีการศึกษา 2564 ลายมือชื่อ อ.ที่ปรึกษาหลัก ..............................................
ลายมือชื่อ อ.ที่ปรึกษาร่วม ..............................................
In recent years, image super-resolution (SR) techniques based on Convolutional Neural Network (CNN) have achieved impressive attention from computer vision scholars and artificial intelligence (AI) companies. Due to the necessity of using the SR algorithms in real-world applications, designing an efficient and lightweight SR algorithm that improves the sharpness and visual quality of the SR results is a critical issue in real-time hardware implementation. To address these issues, we proposed the Multi-FusNet of Cross Channel Network (MFCC) network by constructing the groups of Residual-in-Residual architecture under the multi-path cascading framework. Additionally, a residual connection is used to transfer the low-level features of the early layer to the reconstructed SR image. The proposed SR model is initially trained with the GPU’s training image dataset. To implement our trained model in System on Chip FPGA, the size of the proposed model is required to reduce. We convert the floating-point checkpoint into a fixed-point integer checkpoint in the quantization procedure. According to the experimental results, the proposed method reduces the number of network parameters significantly (8.4 times compared to RCAN), can execute fast in System On Chip FPGA, around 30 frames per second, and image accuracy of the proposed method in terms of PSNR value does not decrease over 1 dB.
ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude to my thesis advisor Assoc. Prof. Dr. Supavadee Aramvith for her continuous support and untiring effort all the time during my study. This achievement would not have been possible without her guidance. I would like to thank my thesis co-advisor Prof. Takao Onoye for giving suggestions, guidance, idea, and viewpoint on my dissertation.

I also would like to thank Chairman Prof. Prasit Prapingmongkolkarn for giving suggestions on my dissertation and giving me a chance of working on the project. I would like to thank the other member of my thesis committee, Asst. Prof. Suree Pumrin, Asst. Prof. Charnchai Pluempitiwiriyawej, and Dr. Sitapa Rujikietgumjorn, for giving critical reviews and advice on my dissertation.

I also appreciate all members of the Video Technology Research Group (VTRG), seniors, and my friends for their support and spending their time for suggestions and guidance.

Last but not least, I would like to express my deepest gratitude to express my family members for their help, unfailing understanding, and affectionate encouragement during my graduate journey.

Watchara Ruangsang
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT (THAI)</td>
<td>iii</td>
</tr>
<tr>
<td>ABSTRACT (ENGLISH)</td>
<td>iv</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>v</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>ix</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>Chapter 1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1. Motivation and Problem Statement</td>
<td>1</td>
</tr>
<tr>
<td>1.2. Objective</td>
<td>3</td>
</tr>
<tr>
<td>1.3. Working Scopes</td>
<td>3</td>
</tr>
<tr>
<td>Chapter 2 Background</td>
<td>4</td>
</tr>
<tr>
<td>2.1. Super-Resolution Algorithm</td>
<td>4</td>
</tr>
<tr>
<td>2.1.1. Single Image Interpolation Based methods</td>
<td>4</td>
</tr>
<tr>
<td>2.1.2. Multiple Image Reconstruction Methods</td>
<td>4</td>
</tr>
<tr>
<td>2.1.3. Learning-Based Methods</td>
<td>5</td>
</tr>
<tr>
<td>2.1.3.1. Classical Sparse Coding Method</td>
<td>6</td>
</tr>
<tr>
<td>2.1.3.2. Anchored Neighborhood Regression Method</td>
<td>6</td>
</tr>
<tr>
<td>2.1.3.3. Self-Exemplars Method</td>
<td>7</td>
</tr>
<tr>
<td>2.1.3.4. Deep Learning Method</td>
<td>7</td>
</tr>
</tbody>
</table>
2.2. Super-Resolution Algorithm in Deep Learning Method........................................8
  2.2.1. Linear Networks..................................................................................................8
  2.2.2. Residual Network...............................................................................................10
  2.2.3. Recursive Networks..........................................................................................17
  2.2.4. Progressive Reconstruction................................................................................20
  2.2.5. Densely Connected Networks............................................................................21
  2.2.6. Multi-Branch Designs ......................................................................................23
  2.2.7. Attention Based Networks................................................................................25
  2.2.8. Multiple Degradation Handling Networks.......................................................26
  2.2.9. Generative Adversarial Nets Network...............................................................27
2.3. Super-Resolution on Hardware ..............................................................................29
  2.3.1. Interpolation based on hardware ......................................................................29
  2.3.2. Multiple frames based on hardware ..................................................................30
  2.3.3. Learning-Based Method ...................................................................................31
  2.3.4. Deep Learning Method ....................................................................................32

Chapter 3 Image Super-Resolution ..............................................................................35
3.1. Multi-FusNet of Cross Channel Network ...............................................................37
  3.1.1. Residual Group: ..............................................................................................39
  3.1.2. Multipath Residual: .........................................................................................40
  3.1.3. Pixel Shuffle Fusion .........................................................................................41
3.2. Dataset Image .........................................................................................................47
  3.2.1. Training Dataset image ....................................................................................47
    3.2.1.1. DIV2K [69] .................................................................................................47
  3.2.2. Testing Dataset Image .....................................................................................47
3.2.2.1. Set5 [70] ........................................................................................................ 47
3.2.2.2. Set14 [71] ....................................................................................................... 48
3.2.2.3. BSD100 [72] .................................................................................................. 48
3.2.2.4. Urban100 [19] .............................................................................................. 49
3.2.2.5. Manga109 [73] ............................................................................................. 49

3.2.3. Testing Dataset Image of HW .......................................................................... 50

3.3. Hardware ............................................................................................................... 50
3.3.1. System on Chip (SoC) FPGA ........................................................................... 50
3.3.2. Hardware Design Flow ..................................................................................... 53

3.4. Performance Evaluate .......................................................................................... 56
3.4.1. Peak Signal to Noise Ratio: PSNR .................................................................... 56
3.4.2. Structural similarity (SSIM) ............................................................................. 56
3.4.3. Frame Per Second (FPS) .................................................................................. 57

Chapter 4 Experimental Results and Discussions ....................................................... 58
4.1. Experimental Setup ............................................................................................... 58
4.2. Result Analysis of Network .................................................................................. 58
4.3. Result Analysis of Hardware ................................................................................ 66

Chapter 5 Conclusion and Future Works ..................................................................... 69

REFERENCES ............................................................................................................. 70

VITA ............................................................................................................................... 79
LIST OF TABLES

Table 1 BASED LINE MODEL AT 50 EPOCH ON SET5 (2×) ................................................. 44
Table 2 THE EXTENSION OF RESIDUAL IN BASED LINE MODEL ..................................... 46
Table 3 RESULTS QUANTITATIVE OF MODEL X2 ................................................................. 59
Table 4 RESULTS QUANTITATIVE OF MODEL X3 ................................................................. 60
Table 5 RESULTS QUANTITATIVE OF MODEL X4 ................................................................. 61
Table 6 RESULTS QUANTITATIVE OF MODEL X8 ................................................................. 62
Table 7 Hardware implementations average PSNR in with a scaling factor 2 of Set5 .................. 67
Table 8 Hardware implementations comparison ..................................................................... 68
LIST OF FIGURES

Figure 1 Example CCTV Image ................................................................. 1

Figure 2 Super-Resolution Convolutional Neural Network PSNR Performance [5]....... 2

Figure 3 Learning a Deep Convolutional Network for Image Super-Resolution
(SRCNN) [21] ................................................................................................. 8

Figure 4 Learning a Deep Convolutional Network for Image Super-Resolution
Network Structure (FSRCNN) [22] ..................................................................... 9

Figure 5 Very Deep Convolutional Networks (VDSR) Network Structure [7] ........... 10

Figure 6 Enhanced Deep Residual Networks (EDSR) for Single Image Super-Resolution
Structure [25] ................................................................................................. 11

Figure 7 Cascading Residual Network (CARN) [32] ............................................. 12

Figure 8 Multi-scale Residual Network for Image Super-Resolution (MSRN) [33]..... 13

Figure 9 Super-Resolution Feedback Network (SRFBN) [34] ................................... 14

Figure 10 Adaptive Weighted Super-Resolution Network (AWSRN) [35] ............... 14

Figure 11 Multi-scale inception-based super-resolution Structure (MSISRD) [36] ...... 15

Figure 12 Lightweight Single-Image Super-Resolution Network with Attentive Auxiliary
Feature Learning (A2F) [37] Structure ............................................................. 16

Figure 13 Multi-objective reinforced evolution in mobile neural architecture search
(MoreMNAS) Architecture [38] ......................................................................... 17

Figure 14 Deeply-Recursive Convolutional Network (DRCN) Structure [8] .......... 18

Figure 15 Deep Recursive Residual Network (DRRN) Structure [40] ...................... 19

Figure 16 Memory Network for Image Restoration (MemNet) Structure [23] .......... 20

Figure 17 Laplacian Pyramid Super-Resolution Network (LapSRN) [6] ............... 21
Figure 18 Deep Residual Dense Network (RDN) Structure [26] .............................................. 22
Figure 19 Deep Back-Projection Networks (DBPN) for super-resolution [27] ....................... 23
Figure 20 Fast and Accurate Single Image Super-Resolution via Information
Distillation Network (IDN) [41] ........................................................................................................... 24
Figure 21 Image Super-Resolution with Information Multi-distillation Network (IMDN)
[41] ....................................................................................................................................................... 25
Figure 22 Deep residual learning for image recognition Network (SelNet) [42] ................. 25
Figure 23 Residual Channel Attention Network (RCAN) Structure ........................................ 26
Figure 24 Super-Resolution Network for Multiple Degradations Structure (SRMDNF)
[24] ......................................................................................................................................................... 27
Figure 25 Generative Adversarial Nets Network Structure [29] ........................................... 28
Figure 26 Interpolation based on Hardware [43] ...................................................................... 29
Figure 27 Interpolation based on Hardware [44] ..................................................................... 30
Figure 28 Multiple Frame based on Hardware [45] ................................................................. 30
Figure 29 Multiple-frame registration on hardware [46] ......................................................... 30
Figure 30 Block Diagram of Learning-Based Method on Hardware [47] ................................. 31
Figure 31 Block Diagram of Learning-Based Method on hardware [48] ................................. 31
Figure 32 Field Programmable Gate Array (FPGA) for real-time Video SR [49] ............... 32
Figure 33 Block Diagram of Deep Learning Method on Hardware [50] ................................. 33
Figure 34 Block Diagram of Deep Learning Method on Hardware [51] ................................. 33
Figure 35 Block Diagram of Deep Learning Method on Hardware [52] ................................. 34
Figure 36 Multi-FusNet of Cross Channel Network for Image Super-Resolution ............ 38
Figure 37 PSNR vs. Parameter ........................................................................................................ 43
Figure 38 Multi-Residual Networks [68] Structure ................................................................. 45
Figure 39 An example of an image in DIV2K image .............................................................. 47
Figure 40 An example of an image in Set5 ................................................................. 47
Figure 41 An example of an image in Set14 ............................................................... 48
Figure 42 An example of an image in BSD100 .......................................................... 48
Figure 43 An example of an image in Urban100 ....................................................... 49
Figure 44 An example of an image in Manga109 ..................................................... 49
Figure 45 Example of an image in Testing Dataset Image of HW ......................... 50
Figure 46 Kria KV260 specifications ........................................................................ 52
Figure 47 Hardware Design Flow ............................................................................. 53
Figure 48 Kria KV260 Model .................................................................................. 54
Figure 49 Qualitative comparison over datasets for scale x4. The red rectangle indicates the area of interest for zooming ................................................................. 63
Figure 50 Qualitative comparison over datasets for scale x4. The red rectangle indicates the area of interest for zooming ................................................................. 64
Figure 51 Qualitative comparison of img093 belongs to Urban100 dataset for scale x8. The yellow rectangle indicates the area of interest for zooming ..................... 65
Figure 52 Qualitative comparison of an image belongs to Manga109 dataset for scale x8. The yellow rectangle indicates the area of interest for zooming ..................... 65
Figure 53 Target Resolution ...................................................................................... 66
Figure 54 Result of KV260 processing super resolution method MFCC 2x ............. 66
Chapter 1
Introduction

1.1. Motivation and Problem Statement

A closed-circuit television (CCTV) is used for urban security and environmental monitoring. In the setup of smart cities, home video surveillance systems provide the necessary visual information to improve situational awareness, enhance security, deter potential theft, and provide remote monitoring. But the output image of CCTV may have low resolution (LR) or loss of information because of the camera setup and lighting [1] as shown in Figure 1.

![Example CCTV Image](image)

*Figure 1 Example CCTV Image*

Face recognition and understanding information such as texts and tasks are applied in other aspects. If a person’s face is taken from a long distance, the output image is hard to detect and learn to recognize faces. The computer algorithm for improving the resolution of the LR images is known as Super-Resolution (SR) [2].

Even though bicubic interpolation [3] is widespread in SR because of its fast speed, the HR image output is not clear and begins to blur. The dictionary method,
also called sparse coding (SC) [4] can produce better quality results than the bicubic interpolation. SC utilizes the relationship between LR and HR images. At present, the most widely used method is the convolution neural network (CNN) [5]. This method is also used to learn from image datasets. The HR image output of the CNN method has a higher peak signal-to-noise ratio (PSNR) value than the bicubic and the sparse coding methods, as shown in Figure 2. Hence, the image quality of using CNN is better than other methods.

![Figure 2](image-url)

*Figure 2 Super-Resolution Convolutional Neural Network PSNR Performance [5]*

In [6-8], graphics processing unit (GPU) processing to process CNN makes them more expensive. Typically, a convolution process uses the whole image. If the method uses patch processing, there is some processing time for the path sequence process. To solve this problem, the advantage of parallel processing in field-programmable gate array (FPGA) can help the repetitive process faster. Moreover, FPGA can design on the chip for some applications. Integration of FPGA with central processing unit (CPU), memory, and input/output port by low power consumption is called System on Chip (SOC). The objective of combined SoC with FPGAS is to use some libraries or processing from CPU with parallel processing of FPGA for faster processing.
In this research, we are interested in HR imaging using SR based on CNN [5]. SR is an algorithm to reconstruct HR images. The challenge is to be able to perform in real-time. An embedded super-resolution approach can be implemented and connected directly to CCTV to solve that problem. This approach enables real-time processing with a faster and more optimized algorithm version. There are several CNN layers to learn the features of the image for reconstruction in a typical CNN-based SR algorithm. This network structure requires a considerable amount of processing time.

1.2. Objective

1. Develop the algorithm for fast image super-resolution based on a convolution neural network.

2. Proposed super-resolution on hardware for fast image super-resolution.

1.3. Working Scopes

1. This work used processing from image frame by frame.

2. The proposed algorithm can process fast super-resolution in System On Chip FPGA at least 30 frames per second.

3. The image accuracy of the proposed method in terms of PSNR value does not decrease over 1 dB.

4. The performance measurement of the algorithm is tested with:

   -> Set5 image dataset (5 images)
   -> Set14 image dataset (14 images)
   -> Urban100 image dataset (100 images)

Next, we will describe the related work in the following section:
Chapter 2
Background

2.1. Super-Resolution Algorithm

The SR approaches of using a single LR image to create an HR image are more accessible and faster than using multiple LR images to create HR images. The multiple images can find more lost data but take exceedingly long processing time and are more complex. The SR method can be divided into three techniques: single image interpolation-based, multiple image reconstruction, and learning-based.

2.1.1. Single Image Interpolation Based methods

Single Image Interpolation methods use a single image to create HR image. The process is simple, but some information may be lost. The details of the output image are not quite the same as the original image. This method can be divided into as follows. Smoothing \[9\] makes the HR image with less or disappear noise. The method aims to eliminate or reduce noise. Examples of smoothing are the Gaussian, Median, and Wiener filters. However, in these processes, some details of the image will be lost. The sharpening process will not increase the contrast ratio. But the process will add the details to the HR image. The advantage is adjusting the contrast and noise does not increase.

Moreover, the interpolation has a process that uses pixels or blocks in the image, such as the nearest neighbor, bilinear interpolation \[10\] and bicubic interpolation \[3\]. These methods will include the smoothing and sharpening process. But the details of the HR image on borders, corners, and surfaces in the picture are unclear.

2.1.2. Multiple Image Reconstruction Methods

Multiple Image Reconstruction Method \[11\] uses multiple LR images. Where the images are taken at different angles of the same scene, each LR image is an image
that is in a different position. If the position of the image is in the same position, the HR image acquired has the same information as the LR image. This method is used for creating the HR image when LR images lose a lot of information. To use this method, the first step is to define the pixel’s position by assigning a one-pixel image reference. But the real LR image is difficult to define the position. Suppose the position of the pixel is an error or wrong position; it has a huge effect on the efficiency and accuracy of the HR image. The second step is to compare the pixel’s position with the pixel in HR images. But the information of LR images changed position without a pattern. It cannot be matched with the pixels of the HR image—the estimated need to have an area of HR images from some parts of LR images. The whole process will improve the image resolution. But in the first step, the image is blurred. Therefore, it is resolved using the procedure to estimate the range in the next step. The last step is improving the HR image’s blurring by the pixel area’s average value. To use multiple LR images for the SR method when the input image has very LR. The HR image will have bad quality. This method needs to consider the special characteristics of the objects, especially the face image, license plate image, and gestures that arise.

2.1.3. Learning-Based Methods

Typically, the image has a lot of details. When the analysis is complex and there are tricky, several research works [12-16] believe the relationships between images in LR and HR images can be learned. The method used image data set to determine the relationship. The process that studies how to get all the information from HR images from LR images is called the Learning-Based Method.

The disadvantage of this method is determining the relationship of LR images compared with the HR images required to study a sample data set. That process uses a lot of processing time. It must be finding a way to improve this process to take less.
Learning-based method for SR will divide the image into patch images. Typically, this method is used to find the same information and use the data sets using pre-processing. Under this learning-based method can find missing data of HR images as well. The related research on the learning-based method is related research work as follows.

2.1.3.1. Classical Sparse Coding Method

The concept of the super-resolution (SR) method is to reconstruct high resolution (HR) image by multiplying LR image $Y$ with blurred image $H$ and down-sampled factor $D$, resulting in the HR image $X$, as shown in (1)

$$Y = DHX$$  

Yang et al. [17] proposed a sparse representation for dictionary coefficient from the trained image data and patch processing to learn the relationship between LR and HR images. As shown in (2), be composed of operator feature extraction $F$, LR dictionary coefficient $D_l$, sparse representation $\alpha$, LR input patch $y$, and weight value $\lambda$. After that, HR is reconstructed from sparse representation with dictionary coefficient. This method can improve resolution quality but has more processing time.

$$\min_{\alpha} \|FD_l \alpha - Fy\|_2^2 + \lambda \|\alpha\|_0$$  

2.1.3.2. Anchored Neighborhood Regression Method

Timofte et al. [18] proposed Anchored Neighbor Regression (ANR) as the fast learning-based method. The method reconstructs HR from LR patches from neighborhoods exemplar using ridge regression ($l_2$-norm) as shown in equation (3). $N_l$ is neighborhood LR image correlation and $\beta$ is weight vector.

$$\min_{\beta} \|IF - N_l \beta\|_2^2 + \lambda \|\beta\|_2$$
2.1.3.3. Self-Exemplars Method

Huang et al. [19] proposed self-exemplars for creating an HR image. These methods are based on learning an image by itself without training data. HR image output used high processing time.

2.1.3.4. Deep Learning Method

In recent years, deep Convolutional Neural Networks (CNN) have been useful for SR with a higher reconstruction quality than the learning-based method. CNN usually has complex structures; thus, the computational time can also be higher. Typically, CNN consists of convolution and deconvolution layers as in (4).

\[ X_i = \max(0, w_i \ast X_{i-1} + b_i) \] (4)

Where \(X_i\) and \(X_{i-1}\) are the output and input signals of an \(i^{th}\) layer. Each layer’s biases and weight values are \(W_i\) and \(b_i\), respectively. The last parameter (*) is the convolution operation. The reconstruction layer represents in (5).

\[ X_i = w_i \ast X_{i-1} + b_i \] (5)
2.2. Super-Resolution Algorithm in Deep Learning Method


2.2.1. Linear Networks

The simple network is Super-Resolution using deep Convolutional Neural Networks (SRCNN) [5, 21] the first state-of-the-art CNN for SR methods. SRCNN is based on CNN, and the HR output achieves high quality in terms of a peak-signal-to-noise ratio (PSNR). The network consists of a feature extraction layer to build an outline input image, a non-linear mapping layer that shows low-resolution features to high-resolution features, and a reconstruction layer to crate HR images from LR patches. The SRCNN uses bicubic interpolation to build an HR image of the same size due to an end-to-end network with Mean Squared Error (MSE) loss function.

![Figure 3 Learning a Deep Convolutional Network for Image Super-Resolution (SRCNN) [21]](image)

After that, Fast-SRCNN (FSRCNN) [22] develops quality with a fast process for real-time applications. The network has four layers of convolution and one deconvolution to an upscale image. The feature extraction layer uses LR images to map features. After that, shrinking is utilized to decrease a very large LR feature
dimension. Expanding is the penultimate step used to achieve high quality in the HR part by adding a layer of a feature. Finally, deconvolution is the important step in creating an HR image with each pixel from the feature value. The idea is to add sixteen recursive layers to improve accuracy. This process adds a layer without a new parameter. The difference with SRCNN is the input size using a smaller filter and extending mapping layers. The approach increases the layer for improved accuracy. Output HR image reconstruction by combining residual learning with bicubic interpolation is shown in Figure 2.

\[ y_k = x_i(m, n) \times w_i \]  

Very Deep Super-Resolution (VDSR) [7] increases the layer for improved accuracy. Output HR image reconstruction by combining residual with bicubic interpolation is shown in Figure 4.
interpolation. That network names from team join competition represent convolution same size to fend off convergent.

Figure 5  Very Deep Convolutional Networks (VDSR) Network Structure [7]

2.2.2. Residual Network

Deep Super-Resolution (EDSR) [25] used residual nets (ResNet) [31] by deleting Batch Normalization layers and ReLU activation to a lower number of parameters. EDSR method exceeds in performance of current SOTA models. The significant performance improvement is due to optimization by removing avoidable studies programs widely used residual networks and expanding the model's size for setting up the training course. It also suggests a new system MDSR and training strategy for reconstructing HR images for different scaling factors in a single model rather than multiple models. This method carefully designs the configuration and uses practiced operations, which are the key requirement for training the meshes. This model solves two problems: first, for training the model for altered scaling aspects in a particular model, and the second issue is the increased computation time and memory. These two problems can be solved by reducing unnecessary modules in SRResNet [29] architecture and using appropriate loss functions while training the model. This method uses scale-independent information, i.e., the training of various up-scale models from previously trained down-scale prototypes. Then
again, it uses a multi-scale architecture that shares parameters across different scales. An enhanced super-resolution algorithm is proposed in this paper.

Improved results have been achieved while compacting the model and abolishing avoidable modules from the current widely used ResNet example. Residual scaling techniques are also employed to train bigger miniatures securely. The suggested SISR miniature overperforms present models and attains SOTA comparable achievements. Additionally, for further cutting down the model size and time utilized in training, there occurs an advancement of a multi-scale (MS) SR framework. In a unified net, this MS model can develop a productive pact with varied scales of SR with scale-subjected calibers and collective main nets. The model construction is concurrent to SRResNet, but the activation function ReLU layers are not included in the exterior of the residual blocks. Also, the network dummy involves a residual scaling panel since it uses 64 feature maps for each convolution panel.

Figure 6 Enhanced Deep Residual Networks (EDSR) for Single Image Super-Resolution Structure [25]

This novel method [32] proposes a cascading mechanism for reducing the model’s weight and improving efficiency. Most SOTA models have larger computations which are not good for user experience. Hence we propose Cascading Residual Network to increase the performance and speed by suppressing the network's computations for real-world applications. CARN uses ResNet architecture
and a cascading mechanism at local and global levels to include features from all layers. Also, the CARN-M model shows the trade-off between the model’s performance and weight. The CARN-M technique can further be applied to video data since many streaming services require storage for high-quality videos. Thus, a design can be made in stores LR videos; after passing through SR, the system produces high-quality videos. The CARN model out-performs all the state-of-art models having less than 5M parameters. It has an efficient computational cost compared to other SOTA models because the basis of comparison is multi-adds (multiply-operations). The size of CARN-M is 1.6MB which is good for use on mobile devices.

![CASCADING RESIDUAL NETWORK (CARN)](image.png)

**Figure 7 Cascading Residual Network (CARN) [32]**

Most of the existing SR models during the reconstruction have the following problems: Poor Feature Utilization; difficulty to adapt with arbitrary scaling factors. A novel multi-scale residual network (MSRN) [33] has been proposed to address the abovementioned issues. -This model can fuse the image features at different scales. It is the first multi-scale module based on a residual structure which is very easy to train -It is a basic skeleton for stratified feature union and is flexible with different up-scaling factors. The model shows superior performance over the SOTA model on various benchmark datasets. Also, this model can achieve competitive results by increasing the number of MSRBs.
In image SR, multiple HR images may often result in identical LR images. To solve this issue, multiple solutions have already been proposed. One is Feedback Network for Image SR [34] in which a feedback block is designed to handle feedback connections and generate high-quality reconstruction results. The feedback manner is achieved by using hidden states in an RNN. The skip connections and depth connections play a vital role in any of the deep CNNs. The Depth connections give more contextual information, whereas the skip connections reduce the gradient vanishing problem of deeper networks. The proposed method further introduces a curriculum learning strategy that helps to learn complex degradation with difficulties in HR reconstruction. This model also uses data augmentation. The main contribution of this model design are:

- Employing a feedback mechanism in which high-level information flows in top-down feedback flow, thereby increasing the reconstruction ability with fewer parameters

- Enriches HR through up and down sampling and dense skip connections.

- Applies curriculum learning strategy to learn complex degradations.

Experimental results show superiority of the proposed model over SOTA models in network parameters with BI degradation model, BD, and DN degradation model.
With the fast-growing techniques in the field of super-resolution, Deep Learning has been successfully introducing many model designs to emend the achievement of SR prototypes. However, most CNN-based networks require heavy computation and thus have limited real-world application. The model Adaptive Weight SR network (AWSRN) [35] is designed to resolve this subject. This model consists of Local Fusion Blocks (LFB) designed with residual learning-based embryonic adaptive voluminous residual units (ARWU) and a local residual fusion entity (LRFU). Apart from LFB, it also contains an adaptive weight multi-scale module (AWMS) for enhancing the reconstruction layer to extract better features. AWMS is an important contributing factor to making the network a lightweight structure. AWSRN consists of distinctive blocks: element abstraction, non-linear mapping, and adaptive weight multi-scale modules. The model shows better results of PSNR and SSIM on std. Benchmark library as compared to SOTA formulas.
Inception Network [36] uses a residual network using asymmetric convolution to reduce the number of parameters. The short and long feature information is extracted using locally residual asymmetric convolutional block and inception-based asymmetric convolutional block architecture by model directly. Compared to the set of SISR models, the MS models remain compact and show competitive results compared to SISR models. The dictated method was evaluated on the standard benchmark dataset and DIV2K dataset. The evaluation was ranked 1st and 2nd in NTIRE SR Challenge. This method is better than the state-of-the-art quantitative and qualitative analysis model for 2, 4, and 8 scaling factors on five benchmark datasets. The model construction is concurrent to SRResNet, but the activation function ReLU layers are not included in the exterior of the residual blocks. Also, the dummy involves a residual scaling panel since it uses 64 feature maps for each convolution panel.

Figure 11 Multi-scale inception-based super-resolution Structure (MSISRD) [36]

The network Lightweight Single Image, SR Network with Attentive Auxiliary Feature Learning [37] shows that model performance can be greatly improved by
using auxiliary features and channel attention since the studies show that the lesser the number of auxiliary features, the lesser the high-frequency information. To check the effectiveness of auxiliary features, we remove the auxiliary feature branch and check the results with only the attention branch. Even though CNNs have boosted the performance of SISR, they have much higher computational costs restricting their practical applications. Thus A2F model for SISR extracts auxiliary features and projects them to a common space. The channel attention mechanism enhances the use of these identified auxiliary features. Thus, the A2F model, which is fast, accurate, and with low parameters, shows comparable results on five benchmark datasets and better visual results. Also, when the parameters are less than 320K, A2F outperforms SOTA methods for all scales, proving its ability to utilize the auxiliary features better.

![Image: Lightweight Single-Image Super-Resolution Network with Attentive Auxiliary Feature Learning (A2F) [37] Structure](image)

**Figure 12** Lightweight Single-Image Super-Resolution Network with Attentive Auxiliary Feature Learning (A2F) [37] Structure

The design of Neural networks for mobile devices with constrained resources is difficult. A new multi-objective-oriented algorithm, MoreMNAS [38] has been proposed to address this issue. This model performs multi-objective NAS using advantages from both reinforcement learning and NSGA-II. It means it harnesses benefits from both evolution algorithm (EA) and reinforced learning (RL). For speeding up the selection, this method constructs a cell-based search allowing mutation. These natural mutations and reinforced controls maintain a delicate balance between exploration and exploitation. Thus, protect the model from degradation during evolution and make better use of learned knowledge. Experiments show comparable outcomes on SOTA methods with fewer FLOPS on benchmark datasets.
This method is not just limited to mobile device applications but also dominates in other domains.

Figure 13 Multi-objective reinforced evolution in mobile neural architecture search (MoreMNAS) Architecture [38]

The FALSR method [39] Fast and Lightweight SR with Neural Architecture Search contributes to maximizing the balance between the image restoration and the models' weight. For the SOTA model, it is difficult to reform the realization of the prototype and constrain the resources. The methods successfully improve the performance of this model with constrained resources using neural architecture search. Usually, in SOTA models, the SR is achieved using a multi-objective approach. Still, in the proposed model, an elastic search approach is used, which is based on a hybrid controller for both micro and macro levels. This elastic search helps reduce the model's computational cost and boosts its capacity. Except for deeper networks such as RDN or RCAN model gives better results than other SOTA methods with comparable FLOPS.

2.2.3. Recursive Networks

Residual learning helps accelerate the learning process as a Deeply Recursive Convolutional Network for Image Super-Resolution (DRCN) [8] with a recurrent block by an individual convolution layer. Moreover, Kim et al. [8] proposed Deeply-Recursive Convolutional Network for Image Super-Resolution (DRCN). The idea is to add sixteen recursive layers to improve accuracy. This process adds a layer without a new parameter.
A Deep Recursive Residual Network [8] consisting of almost 52 convolutional layers is being proposed in this network that strives for a deep yet concise network to be applicable in mobile systems since they use less storage space. It adopts residual and recursive learning. Residual Learning: Global Residual Learning is specifically adopted to remedy the difficulties of training deep networks, whereas Local Residual Learning is used to address the issue of performance degradation in visual recognition and image restoration and for controlling the model parameters while increasing the depth. Every amorphic layer uses LRL, whereas GRL is carried off among LR and HR data. The DRRN substantially exceeds SOTA in SISR while using slighter panoramas; this has been indicated by extensive benchmark evaluation. A residual entity infrastructure is learned iteratively in a residual section in DRRN. Stacking a handful of iterative sections together to prepare the residual data is done. After this, the remaining data is concatenated with the LR data from an identity section for regenerating the HR pictures. This method effectively learns lateral calibration from the LR intake data to the HR destination image. DRRN is a rooted, brief, and preferable example in the field of SISR; benchmark experiments and analysis have extensively proved this.
A persistent memory network for image restoration (MemNet) [23] used a memory block in the network of SRCNN. This paper proposes an extremely rooted reminiscence network called MemNet. It familiarizes a recursive and a gate unit to store the information through an adaptive learning process. Most state-of-art models, such as VDSR and DRCN, use short-term memory since each state is directly linked with its former state. However, some models like ResNet have skip connections, which pass information for many layers. Thus, they are restricted to long-term memory. Deeper networks lack long-term memory. To remove this problem, MemNet has been introduced, which consists of a recollection block having a recursive unit and gate unit. The recursive unit is for storing short-range cognizance. The recollection blocks contain the longstanding cognizance. The information from recollection blocks and the recursive units is concatenated at the gate to maintain a persistent memory. So far, MemNet is the most rooted mesh for image restoration. The MemNet attains SOTA completion in denoising the image, SR, and JPEG unblocking. MemNet shows superiority over state-of-art methods on benchmark datasets.
2.2.4. Progressive Reconstruction

Lai et al. [6] proposed Laplacian Pyramid Super-Resolution Network (LapSRN). The method used a convolution layer to upscale the filter step to obtain the loss function. Then each output scale image result combines the loss. The result is high-quality, but the configuration is a high parameter number. This paper proposes a Laplacian Pyramid Framework (LapSRN) which uses progressive up-sampling to reconstruct fast and accurate residuals of HR images. The drawbacks of previous state-of-art models, such as high computational cost, blurry images, and learning difficulty, are overcome by LapSRN. This method uses cascaded CNNs to predict sub-band surplus in rough to fine texture. The model uses the Charbonnier loss function instead of L2 loss to reduce the prediction of blurry visuals. Instead of using predefined up-sampling methods like Bicubic Interpolation, the method uses progressive up-sampling. LapSRN performs better in terms of accuracy, speed, and progressive reconstruction. The LapSRN method has 27 layers overall and takes LR as input, uses residual learning, and progressive reconstruction with a Charbonnier loss function. LapSRN has two branches in its network architecture: Feature Extraction and Image Reconstruction. The proposed method constructs high-quality HR images at a relatively faster speed than other state-of-art methods. It also helps to remove the blur kernels. The only problem with the model is that it does not hallucinate fine details over large scales.
2.2.5. Densely Connected Networks

A residual connection connects dense connections in the Residual Dense Network (RDN) [26]. Most state-of-the-art methods do not use the information of each layer of the network, which can also be termed hierarchical information. Hierarchical features from deeper networks can give more clues for reconstructing HR images. The proposed method thus addresses these drawbacks and utilizes the information from each subsequent layer, or it could be said it does use the hierarchical features using the dense block with a low growth rate, making the training easier for the network. This model consists of a Residual Dense Block (RDB) acting as a building module for the RDN. RDB further consists of Local Feature Fusion (LFF) and densely connected layers. The output of one RDB is directly fed into the next RDB resulting in a continuous state known as Contiguous Memory (CM). This extracted info is adaptively preserved. It also preserves the hierarchical features extracted by Global Feature Fusion (GFF). This network extracts and adaptively fuses features from all layers in LR space. The Local Feature Fusion (LFF) stabilizes wider networks' training and adaptively controls the information preserved from present and previous RDBs. The Local Residual Learning used in the network improves the
flow of information and gradient. By fusing the features extracted from local and global feature fusions, RDN leads to deep image supervision. The model achieves superior performance over five benchmark datasets compared to state-of-art models.

![Deep Residual Dense Network (RDN) Structure](image)

The network represents flexibility in SR images. Also, Dense Deep Back-Projection Network for super-resolution (D-DBPN) [27] adds feedback from LR and HR images with multiple up and down scale blocks. But the method-made network has high complexity. The proposed Deep Back Projection Networks focuses on the mutual dependencies of the LR and HR images by exploiting up-sampling and down-sampling layers of the network and by providing a feedback mechanism for projecting errors in every stage. Unlike other feed-forward networks, methods use multiple iterative up and down sampling stages and a feedback network to increase the extracted SR features. Network architecture consists of three parts:

1.) Initial feature extraction-This stage reduces the dimension of the LR image to make it suitable for entering the projection stage.,

2.) back-projection stage- This unit alternates the construction between the LR and HR design types in which each unit has access to the output of all previous units.
3.) Reconstruction stage- In this stage, finally, the HR picture is obtained by concatenating the design maps created in each protrusion entity.

The proposed network shows better results on large scaling factor 8 enlargement and five benchmark datasets than other state-of-art methods.

Figure 19 Deep Back-Projection Networks (DBPN) for super-resolution [27]

2.2.6. Multi-Branch Designs

The network is a well-known fact that all the CNN-based models have faced the problem of computation complexity and memory consumption. To solve this drawback, a deep but compact CNN has been proposed. This IDN model [41] consists of distinctive parts: an element abstraction module, linearly shaped information refining, and a representative block. The proposed IDN has multiple DBlocks (enhancement unit + compression unit) for extracting short and long feature maps directly from LR images and generating residual reconstructions. IDN is a concise structure, so it is much better in speed and accuracy than several CNN-based SR methods. The proposed method uses group convolutions and performs favorably great in terms of fast execution and fear number of filters per layer on most datasets. This method also performs great in the Information Fidelity Criterion (IFC) metric, which correlates with the human perception of images.
Recently designed Deep Networks have powerful representational capabilities but excessive computations and cannot be implemented over arbitrary scaling factors. To address these drawbacks, a lightweight information multi-distillation network (IMDN) [41] has been proposed containing selective and distillation parts. The distillation parts extract the hierarchical features, and the fusion module aggregates them. The fast and accurate results of the model are due to the information provided by the multi-distillation block IMDB and the Contrast wave attention layer CCA. The method incorporates an adaptive cropping strategy (ACS), allowing the network to have down sampling operations to work with images of any arbitrary scale. This model can reduce the computational cost, memory occupation, and inference time by using ACS. By using this model, the actual factors affecting the inference time can be interpreted. Experimental evaluations show that the proposed network shows great results in visual quality, speed, and memory consumption; also, this method achieves balance in all the factors affecting practical use.
2.2.7. Attention Based Networks

The SelNet [42] Deep CNN with Selection Units for SR is motivated by the linear mapping techniques used in other CNN architectures. The Rectified Linear Unit (RLU) was used for linearly mapping the LR images, inspired to create a non-linear unit known as Selection Unit (SU). Since SU combines identity mapping and a sigmoid switching function, it has better control over data passed through than ReLU. The SU can better handle the non-linearity functionality and is more flexible than ReLU. Thus, this model gives better qualitative and perceptual results. Results show the proposed network has much lower computational complexity and outperforms baseline only with ReLU and other SOTA DL-based SR methods.

The result is high-quality, but the configuration is a high parameter number. Attention Network Currently, state-of-the-art of SR [20, 30] is Residual Channel
Attention Network (RCAN) [28] utilizes residual in residual (RIR) consisting of Residual Group (RG) and Residual Channel Attention Block (RCAB). The structure RG used a short skip connection network as a residual component, RCAB aspects long skip connection network to target feature component of Low Resolution (LR). Addition channel attention (CA) introduces to affect the feature rescale channel. The complexity of the processing takes time with a higher number of parameters.

![Residual Channel Attention Network (RCAN) Structure](image)

**Figure 23** Residual Channel Attention Network (RCAN) [28] Structure

### 2.2.8. Multiple Degradation Handling Networks

Also, Learning a single convolutional super-resolution network for multiple degradations (SRMDNF) [24] applied noise-free degradations to unique efficiency PSNR. A Convolutional SR network for Multiple Degradation has been proposed to deal with multiple degradation processes of SISR. This network has high flexibility in supervising many deteriorations using a SISR model. This technique mainly focuses on exploiting the reprisal of CNN, such as alacrity, computing, exactitude, and advances in training design networks. While these CNNs give great results for a single simplified degradation process, these CNNs fail to give good results for multiple degradations or if the degradation deviates from a simplified degradation.

To solve these issues, CVPR has been proposed. It gives an elementary, compelling, and rooted CNN frame for single model SR. This model works for simple
bicubic degradation and applies for numerous and spatially dissimilar deteriorations. This technique also gives a stretching strategy to look for dissimilarity in the dimensions of LR input data, unclear kernels, and noise levels. We show that the dictated framework obtained by learning from unnatural image data also gives competitive results for real LR images.

The proposed method takes the concatenation of LR images and degradation maps (i.e., unclear kernels and turbulence levels) as input. It allows a SISR model to work with diverse and spatial degradation processes. The proposed model favors multiple and special variant degradations; evaluations are based on benchmark datasets.

![Figure 24 Super-Resolution Network for Multiple Degradations Structure (SRMDNF) [24]](image)

2.2.9. Generative Adversarial Nets Network

Lastly, Generative Adversarial Nets Network [29] creates SR near-natural image by MSE loss with matric similarity and adversarial loss. Although the outputs of this network have better visual quality, the PSNR values are lower than other methods. This paper presents SRGAN Generative Adversarial Network for SR. This method is used to resolve finer details over large scaling factors. For the other state-of-art models, capturing perceptual differences with great texture details is hard. This framework has been proposed with perceptual loss functions consisting of adversarial and content loss. The adversarial loss uses a discriminator network and
gives results like natural images. The content loss is used, which gives similarity in perception instead of similarity in pixels. Thus, GAN networks are good for generating beautiful natural images with high perceptual quality. The proposed network shows great performance in MOS Testing. This method also shows that the standard qualitative measures like PSNR or SSIM fail to capture the image quality w.r.t the human visual system. The proposed SRGAN show comparable results on public benchmark datasets. It also shows some limitations of qualitative measures of image SR. Finally, it shows great reconstructions of images for large upscaling factors 4.

Figure 25 Generative Adversarial Nets Network Structure [29]
2.3. Super-Resolution on Hardware

The SR research on the hardware can be into three methods, similarly to the previous section.

2.3.1. Interpolation based on hardware

Bowen et al. [43] used the multi-frame interpolation based on the total weight mean SR with an iterative, as shown in Figure 26. The testing was done on the Xilinx Virtex II XC2V6000 FPGA board with 4 MB internal and external ram 8MB HR output with HD (1280x720) at 60 fps. From the experimental results, the HR image quality is not good, and noise occurs.

Lee et al. [44] proposed to upscale the image from Full High Definition (FHD 1920x1080P) to 4K ultra-high-definition (3840x2160P). The system upscales the image using Lagrange interpolation, and the interpolation error improves the sharpening filter, as shown in Figure 27. The operation in the system use adds and shifts for computation. HR image used a processing time of around 30 fps.
2.3.2. Multiple frames based on hardware

Seyid et al. [45] proposed creating HR using multiple frames, as shown in Figure 28. The sub-pixel differences and the iterative back-projection are parallel processing on FPGA. The image HR output size is 512x512 pixels at 25 fps from processing 20 LR images.

Redlich et al. [46] used infrared cameras size 160x120 pixel to create HR size 640x480 pixels using multiple-frame registration. As Figure 29, the pyramid generator reduces the image size to help in registration images, which are processed at 150 FPS since the images are small. The implement board is Xilinx Spartan-6 LX45 FPGA.
2.3.3. Learning-Based Method

Yang et al. [47] proposed an SR system that uses five input line buffers and eight output line buffers for output image FHD size at 60 fps using a dictionary learning-based approach. They used anchored neighborhood regression (ANR) to create output with feature extraction from principal component analysis (PCA), reducing dimension for producing HR patch. As a result, the method used high memory and logic gates in FPGA.

![Block Diagram of Learning-Based Method on Hardware][47]

Kim et al. [48] used application-specific integrated circuit (ASIC) and FPGA to implement with lower memory and gate. The method used two input line buffers. The four output line buffers were used for 4K UHD at 60 fps using an edge orientation in the training step.

![Block Diagram of Learning-Based Method on hardware][48]

This novel design is based on Field Programmable Gate Array (FPGA) for real-time Video SR [49]. The learning-based VSR techniques suffer from the following two problems: - large computational complexity - temporal inconsistency. To solve these
issues, ERVSR exploits the temporal information of LR space in a hidden way and represents HR in the residual framework. It improves model efficiency and reduces the parameters by performing convolutions on LR input and hidden state and introducing channel modulation coefficient. Statistical normalization and fixed-point quantization compress the hidden state and reduce memory consumption. And to reduce the network complexity, separable and group convolutions are adopted. The model ERVSR is evaluated on multiple datasets. Results demonstrate it performs better than SOTA models. The proposed model also helps improve the temporal consistency by using lightweight recurrent architecture.

![Diagram](image)

Figure 32 Field Programmable Gate Array (FPGA) for real-time Video SR [49]

2.3.4. Deep Learning Method

The classification CNNs and the Super Resolution CNNs [50] require a lot of memory access which is not good for the mobile user experience. Even though FPGAs are good replacements, they still face the issue because of large on-chip memory and can produce the throughput of 5fps, which is insufficient for small edge devices. To resolve these issues, three techniques have been classified. a.) Layer fusion based on selective caching, b.) memory compaction, c.) cyclic ring core architecture. The proposed Processor infrastructure consists of a controller, local rings having a convolutional core, weight memory parallel processing engines. Router and an accumulator, all fabricated in CMOS process having 50-200 MHz clock frequency. The designed processor shows 60fps throughput for scale x4 and 25 fps for scale x2, and 60 fps for full high-definition resolution.
Computational complexities are the major prohibition for implementing SR on small edge hardware. To solve this issue, a first method is proposed implementing real-time CNN hardware upscaling 2K FHD video frames to 4K UHD video at 60 fps using FPGA. The proposed method [51] contributes to: Implementing SR on hardware utilizing less memory by processing the input line by line and maintaining the parametric values and proposing a cascaded structure of 1D convolutions for sustaining large horizontal receptive fields while keeping smaller vertical fields. This network also reduces the number of filter parameters by introducing depthwise separable convolution with residual connections and introducing a simplified and effective quantization process to convert 32-bit FP data to fixed point data and a compression method to reduce line memory. The proposed SR-based Hardware can efficiently reconstruct 4K UHD at 60 fps and can be implemented for more than twice up-scaling or arbitrary up-scaling.

To decrease the computational load and enlarge the utilization of the process element (PE) of the CNN-based SR approach, Lee et al. [52] proposed a data
flow for hardware-effective purposes by enlarging data reuse and equally assigning the computational workload for each layer. The proposed data flow calculated the pixel in the receptive field row-wise by arranging memory addresses with a circular shift. To reduce the expensive calculation of a single pipeline step, they utilized the partial convolution-based pipeline architecture instead of the conventional convolution, as shown in Figure 35. To support a 4K UHD video stream with 60 fps by embedding the proposed architecture on an FPGA.

Figure 35 Block Diagram of Deep Learning Method on Hardware [52]
Chapter 3

Image Super-Resolution

The image Super-resolution (SR) aims to reconstruct the high-resolution (HR) image from the low-resolution (LR) input. The SR is considered an ill-posed problem. Therefore, there are many existing SR techniques to produce the HR image. However, the SR image's perceptual quality and the SR model's execution time are two critical factors in designing an effective and robust SR model for applying in real-world applications. A real-world problem in utilizing closed-circuit television (CCTV) [53] for intelligent monitoring [54] such as traffic congestion and accident [55], home management security [56], face recognition [57], and social identification [58] is appeared due to the low-resolution (LR) images produced by the CCTV camera [59].

Many research topics such as feature preservation in video coding [60, 61] and super-resolution [36, 62, 63] have been published to solve the low-resolution problem. Deep Convolutional Neural Networks (CNN) have recently been a powerful model in solving the ill-posed SR problem [30]. Dong et al. [5, 21] proposed Super-resolution using deep Convolutional Neural Networks (SRCNN) model with the architecture of a three-layer network that is linearly stacked together. In this shallow and straightforward architecture, the first layer is designed to extract the features from the LR input, the second layer is used for the non-linear mapping from low-dimension to high-dimension features, and the final layer is responsible for aggregating the previous layer’s feature maps to the final HR result. This SR network is trained in an end-to-end approach to minimize the Mean Squared Error (MSE) between the ground truth (GT) images and the reconstructed (SR) image. Following by SRCNN model, other models, including Very Deep Super-resolution [7] (VDSR) and Deeply Recursive Convolutional Network (DRCN), achieved significant improvements compared to the SRCNN and FSRCNN models by increasing the depth of their CNN architectures.

Following DRCN and VDSR models, various architecture models based on different approaches, including the residual network [31-39], Recursive Network [8,
progressive reconstruction [6], dense connection network [26, 27], multi-branch architecture [41, 64], attention-based [28, 42] mechanism have been proposed to improve the result of SR model by increasing the data flow of information in the deep learning architecture.

Inspired by Residual network [31] (ResNet) architecture, some SR models attempted to design deeper SR architecture. Based on the residual concept, Lim et al. proposed a very deep SR architecture known as Enhanced Deep Residual Network [25] (EDSR). This SR model stacks residual blocks to design deeper networks (almost 165 convolutional layers) and achieved considerable improvement compared to the earlier SR models. However, training such a deep network trainable network like EDSR [25] is challenging, and the numerous network parameters of a very deep network are an important obstacle for fast execution in real-world applications and hardware implementation.

On the other hand, some recent CNN-based models [6, 8, 39, 41, 64] utilized multi-path network architecture to solve the limitations of the deep network. In this approach, rather than using a single-path network, the multi-path of the network operates in parallel. Based on this concept, the depth of the network is decreased while the performance of the SR model is increased. According to the Multi-path structure, Hui et al. proposed Information Distillation Network (IDN) by designing the cascaded network paths operating in parallel. It means the layers of the network do not require to wait for the previous layer calculation. Additionally, the different types of the extracted features of each path are mixed, consequently improving the SR model's operation time and perceptual result. Although the SR models based on the multi-path approach get acceptable performance and execution time, their results archived low PSNR and SSIM.

Additionally, the low-level feature sharing approach is used in some SR models [34, 35, 37, 42] to enhance the low-frequency information flow in the SR network structure. This low-level feature-sharing approach attempts to transfer the low-level features of the early CNN layer to the latest layers and fuse the low and high-level features. Hence this technique improves the reconstruction quality by
enhancing the sharpness of the SR Image. Motivated by this technique, the SR models such as Feedback Network [34] (SRFBN), Adaptive Weighted Super-Resolution Network [35] (AWSRN), SelNet [42], and Attentive Auxiliary Features [37] (A2F) utilized the feature sharing approach. This idea demonstrates more effective enhancement in the lightweight designed SR architecture.

To deal with these issues, an efficient lightweight SR model for image enlargement has been proposed, stating the main contributions as follows,

1. The multi-depth cross-channel network is proposed to obtain the local pixel attention feature from a low-resolution image.
2. The proposed network also investigates the doubling stage of residual identity connection for retrieving the merged features represented by low-level features.
3. This method explores low-level feature sharing to fuse low-level feature information into unsampled features to enhance the reconstruction ability of the model.

3.1. Multi-FusNet of Cross Channel Network

The proposed network architecture is based on a multi-path residual architecture that designs a wider network rather than a deeper one. The proposed network, named Multi-FusNet of Cross Channel Network (MFCC), consists of four main modules, including feature extraction, multi residual group (Multi RG), enlargement, and low-level fusing, as shown in Figure 36. The Multi-RG architecture has been designed by integrating the RCAN [34] with a multi-identical residual link. Due to the increase in the number of multiplicity (possible paths from the input to the output layer) in the proposed architecture, the information flow between Multi-RGs gradually increases and helps to reduce computational complexity.
The pixel shuffle [65] technique is used as an enlargement module in stated architecture. This technique transforms the low-level feature maps into different channels, transforms the shuffling operation on features, and enlarges the features. The first convolution layer is the feature extraction module which feeds the features to the RG blocks and the fusion module. The idea behind the fusion module is to share the low-level features into the up-sampled features of the multi-RG.

The MFCC network is designed according to the cascading approach, where three different paths constitute the proposed multi-path residual architecture, as shown in Figure 36. The Residual Group (RG) block consists of N stacked Residual Channel Attention (RCA) blocks and a short residual skip connection within the block.

As shown in Figure 36, our model designs Residual Group blocks in the first path and one Residual Group block in the second path. The third path of the model aims to share the low-level features of the earlier layer with the up-sampled features of the multi-path residual. The LR input is first fed to the convolutional layer. The
output is then fed to three different paths. The kernel size of each convolutional layer in the proposed network is k×k. Due to many parameters in the RCAN model, it is challenging to use the model in real-time applications such as hardware implementation. The proposed method aims to reduce the number of residual groups (RGs) in our architecture to have a lightweight architecture.

Additionally, a multi-path residual network architecture is incorporated into our model. Since the multi-path residual architecture can improve accuracy and increase the processing speed, it exhibits more efficient performance in lightweight networks compared to the residual architecture. In other words, we design broader lightweight architectures rather than deeper models. The third improvement is to exploit the low-level features of the early CNN layer and share them with the upsampled features of the multi-path residual network. Due to the lack of high-frequency details in the latest layer of CNN networks, the results of most SR models suffer from over-smooth degradation and show weaknesses in extracting sharp attribute details. Consequently, the resulting images are perceptually unpleasant at large scales.

To address these limitations, a pixel shuffling fusion method is used to fuse low-level features from early layers into the upsampled features from the latest layers of the proposed multi-path residual network. The following section describes the residual group block, multi-path residual configuration, and pixel shuffle fusion method.

3.1.1. Residual Group:

In this section, the Residual Group (RG), which is the robust feature extraction of low-resolution, is presented. The proposed Residual Group is constructed by applying identity residual connection at the edge of N sequences of the statistical
Channel Attention network in [28], where $N>0$. The output of RG, $O_{RG}$, can be defined as in Eq. (7).

$$O_{RG} = Y_{RG} + W_{RG}(F_{RCA_N})$$ (7)

Where $Y$ represents the input of RG. $W_{RG}$ denotes weight parameter in Residual Group block, and $F_{RCA_N}$ is the channel-wise feature from Residual Channel Attention block (RCA). The channel-wise feature from RCA can be determined by Eq. (7) and (8).

$$F_{RCA_N} = H_{RCA}^N(F_{RCA_{N-1}})$$ (8)

$$F_{RCA_0} = H_{RCA}^0(I_{RCA})$$ (9)

Where $F_{RCA_N}$ and $F_{RCA_{N-1}}$ denote the output of $N^{th}$ and $(N-1)^{th}$ Residual Channel Attention blocks, respectively. $H_{RCA}$ shows the corresponding operation function of RCA. $F_{RCA_0}$ is the output of the first Residual Channel Attention block. $Y_{RG}$ shows the input of the first Residual Channel Attention block. The Channel Attention (CA) mechanism is a technique that uses the interdependencies among the feature channels. The CA technique leads to more focus on informative features in the SR model and consequently improves the image reconstruction capability of the model. More details of the CA mechanism and the corresponding operation function in the Residual Channel Attention block can be found in RCAN [28].

3.1.2. Multipath Residual:

As demonstrated in Figure 36, the combination of Residual Group blocks under the multipath-residual architecture [66] is used in our model. Based on multipath residual evidence [66, 67], a wider residual architecture significantly improves the accuracy and computation speed of the model compared to a deeper residual architecture. These improvements are related to raising the multiplicity in the wider
residual network. The multiplicity implies the number of possible paths from the input layer to the output layer. A sequence of two Residual Group blocks is utilized in the first path of our model, and one Residual Group block is employed in the second path. Eq.4 defines the multi-path output of our model.

\[ O_{MR} = O^2_{RG} + O^1_{RG} \]  

(10)

Where \( O^2_{RG} \) and \( O^1_{RG} \) denote the output of two Residual Group blocks in the first path and output of a Residual Group block in the second path, respectively. \( O_{MR} \) denotes the output of the proposed multi-path residual architecture.

3.1.3. Pixel Shuffle Fusion

The low-level feature-sharing approach is employed to improve the reconstructed results' sharpness. Since the low-level features of the early layer contain more high-frequency information, sharing them improves the challenging weakness of the SR model to recover the sharper attributes of the lines and edge. At the same time, it preserves our model against over-smoothing degradation. Our model utilizes the pixel shuffle fusion approach to bypass the low-frequency feature of the early layer of the SR network to the up-sampled features.

The proposed model employs the pixel shuffle [65] technique to up-sample the image. Based on the features sharing concept, the features of the early layer also are up-sampled by the pixel shuffle model and fuse with up-sampled features of the multi-path residual network, as demonstrated in Figure 36.

\[ PS_{Fus} = PS_1(W_{MR}(O_{MR})) + PS_2(W_p(F_L)) \]  

(11)

Where, \( PS_1 \) and \( PS_2 \) represent the pixel shuffle up-sampling on a multi-path residual network and pixel shuffle up-sampling on low-level features sharing. \( W_{MR} \) and \( W_p \) demonstrates convolution operations of the multi-path residual output and
up-sampled low-level features. The pixel shuffle mathematically can be shown in (12).

$$PS(U)_{i,j,c} = U_{\left\lfloor \frac{i}{a} \right\rfloor, \left\lfloor \frac{j}{a} \right\rfloor, C \cdot a \cdot \text{mod}(i, a) + C \cdot a \cdot \text{mod}(i, a) + c}$$ (12)

Where $PS(U)$ is output, $a$ demonstrates scale factor, $i, j$, show pixel coordinates. $C$ is the position of the channel. To modernize the final SR image, the up-sampled high-frequency details of the early layer are fused with the up-sampled features of the multi-path residual model.

$$SR = W_{SR}(PS_{Fus})$$ (13)

Where $PS_{Fus}$ denotes the fusing pixel shuffle result, $W_{SR}$ defines the last convolution operation to produce the final SR result. Utilizing the proposed fused approach improves the capability of our model to recover the sharp attributes of images and improves the perceptual quality of results by preventing an over-smoothing problem.

$$L_1(\emptyset) = \frac{1}{n \times m} \sum_{i=1}^{n} \sum_{j=1}^{m} \|SR(i,j) - y(i,j)\|$$ (14)

Where $SR$ and $y$ demonstrate the result and the reference image, $n$ and $m$ are parameters related to the training dataset.
We have varied the change of RCAB and RG from (20,10) to (1,1). We noticed that the parameter's curve would be saturated from RCAB 5, RG 5 to RCAB 1, RG1. It means that the parameter will not be able to reduce much by decreasing both layers RCAB and RG under five layers. Our target is to obtain the minimal parameter and obtain high PSNR. In terms of this characteristic, we consider the curve’s median the optimal point we could use as the selected parameters. We have computed the median of the existing data. We found that the suitable number of RCAB and RG is RCAB 20 RG 1. Then, we select RCAB 20 and RG 1 as our final architecture.

Figure 37 PSNR vs. Parameter

We have varied the change of RCAB and RG from (20,10) to (1,1). We noticed that the parameter’s curve would be saturated from RCAB 5, RG 5 to RCAB 1, RG1. It means that the parameter will not be able to reduce much by decreasing both layers RCAB and RG under five layers. Our target is to obtain the minimal parameter and obtain high PSNR. In terms of this characteristic, we consider the curve’s median the optimal point we could use as the selected parameters. We have computed the median of the existing data. We found that the suitable number of RCAB and RG is RCAB 20 RG 1. Then, we select RCAB 20 and RG 1 as our final architecture.
Table 1 BASED LINE MODEL AT 50 EPOCH ON SET5 (2x)

<table>
<thead>
<tr>
<th>Model</th>
<th>PSNR</th>
<th>Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCAB (20) in Residual Group (10)</td>
<td>37.90</td>
<td>15,444,667</td>
</tr>
<tr>
<td>RCAB (20) in Residual Group (5)</td>
<td>37.88</td>
<td>7,816,427</td>
</tr>
<tr>
<td>RCAB (20) in Residual Group (2)</td>
<td>37.82</td>
<td>3,239,483</td>
</tr>
<tr>
<td>RCAB (20) in Residual Group (1)</td>
<td>37.72</td>
<td>1,713,835</td>
</tr>
<tr>
<td>RCAB (15) in Residual Group (10)</td>
<td>37.89</td>
<td>11,722,867</td>
</tr>
<tr>
<td>RCAB (15) in Residual Group (5)</td>
<td>37.85</td>
<td>5,955,527</td>
</tr>
<tr>
<td>RCAB (15) in Residual Group (2)</td>
<td>37.75</td>
<td>2,495,123</td>
</tr>
<tr>
<td>RCAB (15) in Residual Group (1)</td>
<td>37.70</td>
<td>1,341,655</td>
</tr>
<tr>
<td>RCAB (10) in Residual Group (10)</td>
<td>37.87</td>
<td>8,001,067</td>
</tr>
<tr>
<td>RCAB (10) in Residual Group (5)</td>
<td>37.80</td>
<td>4,094,627</td>
</tr>
<tr>
<td>RCAB (10) in Residual Group (2)</td>
<td>37.70</td>
<td>1,750,763</td>
</tr>
<tr>
<td>RCAB (10) in Residual Group (1)</td>
<td>37.60</td>
<td>969,475</td>
</tr>
<tr>
<td>RCAB (5) in Residual Group (10)</td>
<td>37.80</td>
<td>4,279,267</td>
</tr>
<tr>
<td>RCAB (5) in Residual Group (5)</td>
<td>37.72</td>
<td>1,415,511</td>
</tr>
<tr>
<td>RCAB (5) in Residual Group (2)</td>
<td>37.61</td>
<td>1,006,403</td>
</tr>
<tr>
<td>RCAB (5) in Residual Group (1)</td>
<td>37.43</td>
<td>597,295</td>
</tr>
<tr>
<td>RCAB (2) in Residual Group (10)</td>
<td>37.70</td>
<td>2,046,187</td>
</tr>
<tr>
<td>RCAB (2) in Residual Group (5)</td>
<td>37.58</td>
<td>1,117,187</td>
</tr>
<tr>
<td>RCAB (2) in Residual Group (2)</td>
<td>37.42</td>
<td>559,787</td>
</tr>
<tr>
<td>RCAB (2) in Residual Group (1)</td>
<td>37.15</td>
<td>373,987</td>
</tr>
<tr>
<td>RCAB (1) in Residual Group (10)</td>
<td>37.61</td>
<td>1,301,827</td>
</tr>
<tr>
<td>RCAB (1) in Residual Group (5)</td>
<td>37.52</td>
<td>745,007</td>
</tr>
<tr>
<td>RCAB (1) in Residual Group (2)</td>
<td>37.17</td>
<td>410,915</td>
</tr>
<tr>
<td>RCAB (1) in Residual Group (1)</td>
<td>36.78</td>
<td>597,295</td>
</tr>
</tbody>
</table>

In addition, by checking the effectiveness of RCAB and RG, we found that if we reduce the layer of RCAB, the performance of PSNR is dropped faster than
decreasing the layer of RG. Moreover, by reducing the layer of RCAB, the performance of the parameter is dropped less than decreasing the layer of RG. Based on this consideration, we chose the RCAB as 20, and RG as 1.

Multi-Residual Networks [68] work residual blocks by expanding residual information to a broader network model. The network with the design of a residual network is called a multi-Residual network. This network can improve the processing speed by removing some parts of the sequential block from a parallel block. It is also noted that the results show better accuracy.

![Multi-Residual Networks [68] Structure](image)

*Figure 38 Multi-Residual Networks [68] Structure*

Multi Residual Based on the experimental results shown, reducing the number of RCABs considerably affects the quality. Since the PSNR value of the baseline model RCAN (RCAB (20) in Residual Group (10)) is 37.513 dB, the goal of the proposed method is to keep that quality in terms of PSNR and reduce the number of parameters and processing time. An experiment is conducted by training the model with 50 epochs using the DIV2K dataset and testing the model with Set5 in scale 2. According to the observation, RG has a negligible impact on quality. Therefore, only one RG block is decided to be used to reduce the number of parameters. On the other hand, if the number of RCABs is reduced, there is a considerable drop in PSNR value. Based on the experimental results, (RCAB (20) in Residual Group (1) is chosen as the final design.
**Table 2** THE EXTENSION OF RESIDUAL IN BASED LINE MODEL

<table>
<thead>
<tr>
<th>Method</th>
<th>Params</th>
<th>PSNR (dB)</th>
<th>Best Epoch</th>
<th>All Epoch</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCAN (Re-Train)</td>
<td>15,444,667</td>
<td>38.25</td>
<td>524</td>
<td>600</td>
</tr>
<tr>
<td>2 Multi Residual Network of branch RCAB (20) in Residual Group (1)</td>
<td>1,713,835</td>
<td>38.06</td>
<td>566</td>
<td>600</td>
</tr>
<tr>
<td>RCAB (20) in 2 Residual Network of Residual branch Group (1)</td>
<td>1,713,835</td>
<td>38.14</td>
<td>576</td>
<td>600</td>
</tr>
<tr>
<td>2 Multi Residual Network of branch RCAB (10) in Residual Group (2)</td>
<td>1,750,763</td>
<td>38.07</td>
<td>543</td>
<td>600</td>
</tr>
<tr>
<td>RCAB (20) in 2 Residual Network of Residual branch Group (1) - Upscale</td>
<td>1,861,547</td>
<td>38.16</td>
<td>417</td>
<td>600</td>
</tr>
</tbody>
</table>

shows the simulation results of several configurations of RCAB and RG with 600 epochs. It is noticed that both the PSNR value and some parameters of (2 Multi Residual Network of branch RCAB (20) in Residual Group (1)) is almost the same as that of (RCAB (20) in 2 Residual Network of Residual branch Group (1)). Another configuration is adding an upscale layer to the network in a multi-path fashion. The edge of the image can be seen more clearly by using this configuration. Moreover, this configuration can achieve a stable model of about 100 epochs faster than other configurations. Hence, this configuration is chosen to be used in the proposed architecture even though the number of parameters and PSNR values are not the best.
3.2. Dataset Image

3.2.1. Training Dataset Image

3.2.1.1. DIV2K [69]

An image database was used in the NTIRE (New Trends in Image Restoration and Enhancement workshop) competition. The images are at 2K resolution, divided into a database for training 800 models and reviewing and testing 100 models.

![DIV2K Image Database](image)

Figure 39 An example of an image in DIV2K image

3.2.2. Testing Dataset Image

3.2.2.1. Set5 [70]

A primary image database is used in research in the field of ultra-high resolution image recovery, which consisted of 5 test images, namely a child’s face, bird, butterfly, head, and woman, at a high-resolution size between 250x361 pixels and 720x576.

![Set5 Image Database](image)

Figure 40 An example of an image in Set5
3.2.2.2. Set14 [71]

There will be more images than in the Set5 image database. However, the image resolution will be low, between 250x361 pixels and 720x576 pixels.

![Example images from Set14](softmax.png, barbara.png, bridge.png, coastguard.png, comic.png, face.png, flowers.png, foreman.png, lenna.png, mam.png, monarch.png, pepper.png, pg3.png, zebra.png)

Figure 41 An example of an image in Set14

3.2.2.3. BSD100 [72]

A primary image database contains 100 test images with resolutions ranging from 321x421 pixels to 481x321 pixels, spanning various images from nature to objects such as airplanes, people, food, and others.

![Example images from BSD100](Figure 42 An example of an image in BSD100)
3.2.2.4. Urban100 [19]

It has the same images as BSD100 but focuses on photographing buildings. with resolutions ranging from 567x1024 pixel to 1280x963 pixel.

Figure 43 An example of an image in Urban100

3.2.2.5. Manga109 [73]

It is the latest database used to test ultra-high resolution image recovery algorithms. The image data set consisted of 109 images for testing from various comic books Made by Japanese artists between 1970 and 2010 with a resolution of 742x1072 pixels to 827x1170 pixels.

Figure 44 An example of an image in Manga109
3.2.3. Testing Dataset Image of HW

The Internet source dataset demonstrates the proposed algorithm’s real-time implementation. The selected image dataset includes ten wallpaper images with a resolution of 640x360, which are reasonably selected for testing in the proposed hardware.

![Example of an image in Testing Dataset Image of HW](1.jpg)

Figure 45 Example of an image in Testing Dataset Image of HW

3.3. Hardware

3.3.1. System on Chip (SoC) FPGA

The central processing unit (CPU) is typically designed from fix gate and core. The processing is based on task and fixed arithmetic engines. Also, the input and output port is fixed. The integrated circuit is also known as System on Chip (SOC) [74]. The circuit consists of memory, CPU, and input/output port with low power consumption. The objective of combining SoC with FPGAS is to use some library or processing from CPU with Parallel processing of FPGA for fast processing. FPGA Field Programmable
Gate Array is programmable logic interconnected digital circuits. They are used because of their higher efficiency and multitasking capacity, which is even better than many microcontrollers and GPUs. FPGAs are easier to manufacture and are more adaptable than Application Specific ICs (ASIC). Mostly used CNN or DNN-based algorithms require efficient hardware architectures. The main drawback in GPU hardware is high power consumption, and for ASIC, it is less adaptable. FPGAs have been employed to address these specific issues, capable of processing billions of 32-bit Floating Point logical and arithmetic operations with fast speed and better accuracy. To increase the processing of convolution operations, Block Random Access Memories are used. The following hardware and framework requirements have been stated for accelerating the hardware solution of ML/DL algorithms and showing the platform's flexibility in incorporating FPGA on adaptive SoC with ARM-based embedded CPU.

We use SoC, which is the combination of CPU and FPGA. For edge-based AI applications, the framework is targeted on Xilinx Kria SoM, which embeds extra port connectivity giving high computation performance based on ZYNQ Ultrascale and MPSoC. For workflow interface, deploying a pre-trained model to Kria SoM is like using a GPU with CUDA and DNN Kit. FPGA was used for workflow demonstration with performance collecting metrics amended by Kria SoM. This demonstration showed good computation evaluation in a low power budget. It is a qualitative analysis to check the abilities of an adaptable platform.
### Kria KV260 specifications

<table>
<thead>
<tr>
<th>Parameter</th>
<th>KV260</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>Zynq UltraScale+ MPSoC</td>
</tr>
<tr>
<td>Form factor</td>
<td>SOM + Carrier Card + Thermal Solution</td>
</tr>
<tr>
<td>Starter kit dimensions</td>
<td>119mm x 140mm x 36mm</td>
</tr>
<tr>
<td>Thermal cooling solution</td>
<td>Active (Fan + Heatsink)</td>
</tr>
<tr>
<td>System logic cells</td>
<td>256K</td>
</tr>
<tr>
<td>Block RAM blocks</td>
<td>144</td>
</tr>
<tr>
<td>UltraRAM blocks</td>
<td>64</td>
</tr>
<tr>
<td>DSP slices</td>
<td>1.2K</td>
</tr>
<tr>
<td>Ethernet interface</td>
<td>One 10/100/1000 Mb/s</td>
</tr>
<tr>
<td>DDR memory</td>
<td>4GB (4 x 512Mb x 16 bit) [non-EDC]</td>
</tr>
<tr>
<td>Primary boot memory</td>
<td>512MB QSPI</td>
</tr>
<tr>
<td>Secondary boot memory</td>
<td>SDHC card</td>
</tr>
<tr>
<td>Device Security</td>
<td>Zynq UltraScale+ MPSoC hardware root of trust (RoT) in support of secure boot. Infineon TPM2.0 in support of measured boot.</td>
</tr>
<tr>
<td>Image sensor processor</td>
<td>OnSiemi AP1302 ISP</td>
</tr>
<tr>
<td>IAS MIPI sensor interfaces</td>
<td>x2</td>
</tr>
<tr>
<td>Raspberry Pi camera interface</td>
<td>x1</td>
</tr>
<tr>
<td>Pmod 12-pin interface</td>
<td>x1</td>
</tr>
<tr>
<td>USB3.0/2.0 interface</td>
<td>x4</td>
</tr>
<tr>
<td>DisplayPort 1.2a</td>
<td>x1</td>
</tr>
<tr>
<td>HDMI 1.4</td>
<td>x1</td>
</tr>
</tbody>
</table>

**Figure 46: Kria KV260 specifications**
3.3.2. Hardware Design Flow

We used Vitis AI to support the latest framework, TensorFlow, and PyTorch for AI. It is an open-source platform that compiles the flow of the model from cloud to edge. Using a Vitis-based AI environment accelerates the interpretation of AI on the Xilinx hardware platform working for both cloud and edge. It is very efficient and user-friendly. Using this platform can develop many deep learning algorithms. Vitis Artificial Intelligence design is implemented with the below-mentioned stature:

![Figure 47 Hardware Design Flow](image)

The proposed SR model is initially trained with the GPU's training image dataset. To implement our trained model in hardware (DPU), we quantize it. We convert the floating-point checkpoint into a fixed-point integer checkpoint in the quantization procedure. Once the quantization procedure is done, the quantized model can be
found in the DPU module. Then the compile operation is applied to the quantized model. The compile.sh shell script is used to compile our quantized model. It creates a .xmodel file that contains the instructions and data. These instructions and data make our model ready to execute on the DPU.

![Kria KV260 Model](image)

**Figure 48 Kria KV260 Model**

For further compaction of model design and higher performance, quantization is used. A quantized model utilizes 8-bit integer tensors instead of 32-bit floating operations. 8-bit Integer quantization is advantageous over 32-bit FLOPs because it reduces the model size and memory bandwidth requirement by up to 4 times. It also provides 2-4 times faster calculation speed than FP32 tensors. The INT8 quantized tensors are particularly used to accelerate the inference and forward calculative operations. The specialty of PyTorch is that it uses FP32 for training and then converts to INT8. All the calculations (forward or reverse) are performed through a fake quantization module using floating-point numbers, and then the trained FP trained model is converted to lower accuracy.

At this level, the low accuracy quantized tensor directly builds the model. This raises a question about what a quantized tensor is and how to generate a quantized tensor in PyTorch. A quantized tensor is a tensor that can store quantized data (i.e.,
INT8/UNIT8/INT32 data) and quantization parameters (i.e., scaling zeros and quantized zeros). The first thing to quantify a tensor in PyTorch is to represent the quantized data as a tensor. This generated quantized tensor can be serially and parallelly connected to perform multiple operations. Two types of processes are used to quantize the tensor, i.e., per-tensor asymmetric linear quantization and per-channel asymmetric linear quantization. In per-tensor asymmetric linear quantization, all the tensors are scaled similarly. In contrast, in per-channel asymmetric linear quantization, each slice of channel dimension (for a particular dimension) of the tensor uses different scaling and offset to reduce the error in the quantization process. This process of quantization is represented through a mathematical mapping equation stated as:

Quantization aware training: This type of quantization is used for acquiring more accuracy where the computations are performed with FP32 and then converted to INT8 quantization. Thus, the above processes compact the model design, achieve higher performance and make the model suitable for mobile edge devices.
3.4. Performance Evaluate

3.4.1. Peak Signal to Noise Ratio: PSNR

We use the measurement Root Mean Square Error: RMSE: A small value indicates high accuracy.

\[
RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (I - T)^2}
\]

By \( I \) is the original image.

\( T \) is an image obtained from various methods.

\( N \) is the image size (width x length).

The measure of the ratio of the signal to noise (Peak Signal to Noise Ratio: PSNR), which is higher, means that the picture has less noise. It can be calculated as follows:

\[
PSNR = 10 \log_{10} \left( \frac{2^n - 1}{RMSE} \right)
\]

where \( n \) is the number of bits used to represent the color value in each image point.

3.4.2. Structural similarity (SSIM)

There are three terms to define the SSIM index: luminance, contrast, and structural need to do a multiplicative combination.

\[
SSIM(x, y) = [l(x, y)]^a \cdot [c(x, y)]^b \cdot [s(x, y)]^r
\]

Where,

\[
l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}
\]
\[
c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}
\]
\[
s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}
\]

(17)
Where $\mu_x, \mu_y, \sigma_x, \sigma_y, \sigma_{xy}$ are the local means, standard deviations, and cross-covariance for images $x, y$. If $\alpha = \beta = \gamma = 1$, and $C_3 = \frac{C_2}{2}$

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

(19)

3.4.3. Frame Per Second (FPS)

A simple and effective performance metric of a deep learning-based SR algorithm is to measure the number of image frames per second, called frame per second (FPS), as Equation (20) is depicted below.

$$FPS = \frac{\text{Total image}}{\text{Time}_{\text{Start}} - \text{Time}_{\text{End}}}$$

(21)
Chapter 4
Experimental Results and Discussions

4.1. Experimental Setup

For the training set of our experiments, we utilize the DIV2K [69] dataset, including 800 images. For the testing set, five standard benchmark datasets such as Set5 [69], Set14 [71], B100 [19], Urban100, and Manga109 are reasonably considered. For degradation models, we apply Bicubic (BI) in our experiments. Y-PSNR, and Y-SSIM are efficiently presented for the performance metrics to evaluate the SR results. For data augmentation, the 800 training images are randomly applied with three rotations such as 90, 180, 120 degrees, and horizontal flipping. We extract 16 in 48×48 test LR color patches to get the input for each training batch. We train our model with an ADAM optimizer, and the parameter values of $\beta_1$, $\beta_2$, and $\epsilon$ are 0.9, 0.999, and $10^{-8}$, respectively. Initially, we set the learning rate as $10^{-4}$. Then, for every $2 \times 10^{-5}$ iterations, we reduce the learning rate to half. Our efficient models are implemented by effectively utilizing PyTorch on Titan Xp GPU.

4.2. Result Analysis of Network

To analyze the performance of our model, several state-of-the-art approaches are chosen. Table 5 describes the parameters for each SR model and Y-PSNR and Y-SSIM values for five datasets with upscale factors 2x, 3x, and 4x, respectively. According to Table 3, we can conclude that our model can achieve higher performance with a reasonable number of parameters than others. For Set 5, Set 14, B100, Urban100, and Manga109, our model improves 0.05 dB, 0.07 dB, 0.02 dB, 0.16 dB, and 0.16 dB over the second-best PSNR model called AWSRN, A2F-L, AWSRN, AWSRN, and A2F-L, respectively, under a comparable number of parameters. As we can see from Tables 5 and 6, our model can achieve a big PSNR improvement of 0.3
dB and 0.26 dB for upscale factors 3x and 4x over AWSRN and A2F-L on the Manga109 data set, respectively.

Table 3 RESULTS QUANTITATIVE OF MODEL X2

<table>
<thead>
<tr>
<th>Model</th>
<th>Param</th>
<th>Set5</th>
<th>Set14</th>
<th>B100</th>
<th>Urban100</th>
<th>Manga109</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSRCNN [22]</td>
<td>12K</td>
<td>37.00/0.9558</td>
<td>32.63/0.9088</td>
<td>31.53/0.8920</td>
<td>29.88/0.9020</td>
<td>36.67/0.9694</td>
</tr>
<tr>
<td>SRCNN [21]</td>
<td>57K</td>
<td>36.66/0.9542</td>
<td>32.42/0.9063</td>
<td>31.36/0.8879</td>
<td>29.50/0.8946</td>
<td>35.74/0.9661</td>
</tr>
<tr>
<td>DRRN [40]</td>
<td>297K</td>
<td>37.74/0.9591</td>
<td>33.23/0.9136</td>
<td>32.05/0.8973</td>
<td>31.23/0.9188</td>
<td>37.92/0.9760</td>
</tr>
<tr>
<td>A2F-SD [37]</td>
<td>313K</td>
<td>37.91/0.9602</td>
<td>33.45/0.9164</td>
<td>32.08/0.8986</td>
<td>31.79/0.9246</td>
<td>38.52/0.9767</td>
</tr>
<tr>
<td>A2F-S [37]</td>
<td>320K</td>
<td>37.79/0.9597</td>
<td>33.32/0.9152</td>
<td>31.99/0.8972</td>
<td>31.44/0.9211</td>
<td>38.11/0.9757</td>
</tr>
<tr>
<td>FALSR-B [39]</td>
<td>326K</td>
<td>37.61/0.9585</td>
<td>33.29/0.9143</td>
<td>31.97/0.8967</td>
<td>31.28/0.9191</td>
<td>38.20/0.9762</td>
</tr>
<tr>
<td>AWSRN-SD [35]</td>
<td>348K</td>
<td>37.86/0.9600</td>
<td>33.41/0.9161</td>
<td>32.07/0.8984</td>
<td>31.67/0.9237</td>
<td>-</td>
</tr>
<tr>
<td>AWSRN-S [35]</td>
<td>397K</td>
<td>37.75/0.9596</td>
<td>33.31/0.9151</td>
<td>32.00/0.8974</td>
<td>31.39/0.9207</td>
<td>37.90/0.9755</td>
</tr>
<tr>
<td>FALSR-C [39]</td>
<td>408K</td>
<td>37.74/0.9586</td>
<td>33.26/0.9140</td>
<td>31.96/0.8965</td>
<td>31.24/0.9187</td>
<td>-</td>
</tr>
<tr>
<td>CARN-M [32]</td>
<td>412K</td>
<td>37.53/0.9583</td>
<td>33.26/0.9141</td>
<td>31.92/0.8960</td>
<td>31.23/0.9193</td>
<td>-</td>
</tr>
<tr>
<td>SRFBN-S [34]</td>
<td>483K</td>
<td>37.78/0.9597</td>
<td>33.35/0.9156</td>
<td>32.00/0.8970</td>
<td>31.41/0.9207</td>
<td>38.06/0.9757</td>
</tr>
<tr>
<td>IDN [41]</td>
<td>552K</td>
<td>37.83/0.9600</td>
<td>33.30/0.9148</td>
<td>32.08/0.8985</td>
<td>31.27/0.9196</td>
<td>-</td>
</tr>
<tr>
<td>VDSR [7]</td>
<td>665K</td>
<td>37.53/0.9587</td>
<td>33.03/0.9124</td>
<td>31.90/0.8960</td>
<td>30.76/0.9140</td>
<td>37.22/0.9729</td>
</tr>
<tr>
<td>MemNet [23]</td>
<td>677K</td>
<td>37.78/0.9597</td>
<td>33.28/0.9142</td>
<td>32.08/0.8978</td>
<td>31.31/0.9195</td>
<td>37.27/0.9740</td>
</tr>
<tr>
<td>IMDN [64]</td>
<td>694K</td>
<td>38.00/0.9605</td>
<td>33.63/0.9177</td>
<td>32.19/0.8966</td>
<td>32.17/0.9283</td>
<td>38.88/0.9774</td>
</tr>
<tr>
<td>LapSRN [6]</td>
<td>813K</td>
<td>37.52/0.9590</td>
<td>33.08/0.9130</td>
<td>31.80/0.8950</td>
<td>30.41/0.9100</td>
<td>-</td>
</tr>
<tr>
<td>SelNet [42]</td>
<td>974K</td>
<td>37.89/0.9598</td>
<td>33.61/0.9160</td>
<td>32.08/0.8984</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>A2F-M [37]</td>
<td>999K</td>
<td>38.04/0.9607</td>
<td>33.67/0.9184</td>
<td>32.18/0.8996</td>
<td>32.27/0.9294</td>
<td>38.87/0.9774</td>
</tr>
<tr>
<td>FALSR-A [39]</td>
<td>1021K</td>
<td>37.82/0.9595</td>
<td>33.55/0.9168</td>
<td>32.12/0.8987</td>
<td>31.93/0.9256</td>
<td>-</td>
</tr>
<tr>
<td>MoreMNAS-A [38]</td>
<td>1039K</td>
<td>37.63/0.9584</td>
<td>33.23/0.9138</td>
<td>31.95/0.8961</td>
<td>31.24/0.9187</td>
<td>-</td>
</tr>
<tr>
<td>AWSRN-M [35]</td>
<td>1063K</td>
<td>38.04/0.9605</td>
<td>33.66/0.9181</td>
<td>32.21/0.9000</td>
<td>32.23/0.9294</td>
<td>38.66/0.9772</td>
</tr>
<tr>
<td>A2F-L [37]</td>
<td>1363K</td>
<td>38.09/0.9607</td>
<td>33.78/0.9192</td>
<td>32.23/0.9002</td>
<td>32.46/0.9313</td>
<td>38.95/0.9772</td>
</tr>
<tr>
<td>AWSRN [35]</td>
<td>1397K</td>
<td>38.11/0.9608</td>
<td>33.78/0.9189</td>
<td>32.26/0.9006</td>
<td>32.49/0.9316</td>
<td>38.87/0.9776</td>
</tr>
<tr>
<td>SRMDNF [24]</td>
<td>1513K</td>
<td>37.79/0.9600</td>
<td>33.32/0.9150</td>
<td>32.05/0.8980</td>
<td>31.33/0.9200</td>
<td>-</td>
</tr>
<tr>
<td>CARN [32]</td>
<td>1592K</td>
<td>37.76/0.9590</td>
<td>33.52/0.9166</td>
<td>32.09/0.8978</td>
<td>31.92/0.9256</td>
<td>-</td>
</tr>
<tr>
<td>DRCN [8]</td>
<td>1774K</td>
<td>37.63/0.9588</td>
<td>33.04/0.9118</td>
<td>31.85/0.8942</td>
<td>30.75/0.9133</td>
<td>37.63/0.9723</td>
</tr>
<tr>
<td>MFCC (Ours)</td>
<td>1861K</td>
<td>38.16/0.9611</td>
<td>33.85/0.9195</td>
<td>32.28/0.9010</td>
<td>32.65/0.9331</td>
<td>39.11/0.9780</td>
</tr>
<tr>
<td>MSRN [33]</td>
<td>5930K</td>
<td>38.08/0.9607</td>
<td>33.70/0.9186</td>
<td>32.23/0.9002</td>
<td>32.29/0.9303</td>
<td>38.69/0.9772</td>
</tr>
</tbody>
</table>
Table 4 RESULTS QUANTITATIVE OF MODEL X3

<table>
<thead>
<tr>
<th>Model</th>
<th>Param</th>
<th>Set5</th>
<th>Set14</th>
<th>B100</th>
<th>Urban100</th>
<th>Manga109</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSRCNN [22]</td>
<td>12K</td>
<td>33.16/0.9140</td>
<td>29.43/0.8242</td>
<td>28.53/0.7910</td>
<td>26.43/0.8080</td>
<td>30.98/0.9212</td>
</tr>
<tr>
<td>SRCNN [21]</td>
<td>57K</td>
<td>32.75/0.9090</td>
<td>29.28/0.8209</td>
<td>28.41/0.7863</td>
<td>26.24/0.7989</td>
<td>30.59/0.9107</td>
</tr>
<tr>
<td>DRRN [40]</td>
<td>297K</td>
<td>34.03/0.9244</td>
<td>29.96/0.8349</td>
<td>28.95/0.8004</td>
<td>27.53/0.8378</td>
<td>32.74/0.9390</td>
</tr>
<tr>
<td>A2F-SD [37]</td>
<td>316K</td>
<td>34.23/0.9259</td>
<td>30.22/0.8395</td>
<td>29.01/0.8028</td>
<td>27.91/0.8465</td>
<td>33.29/0.9424</td>
</tr>
<tr>
<td>A2F-S [37]</td>
<td>324K</td>
<td>34.06/0.9241</td>
<td>30.08/0.8370</td>
<td>28.92/0.8006</td>
<td>27.57/0.8392</td>
<td>32.86/0.9394</td>
</tr>
<tr>
<td>AWSRN-SD [35]</td>
<td>388K</td>
<td>34.18/0.9273</td>
<td>30.21/0.8398</td>
<td>28.99/0.8027</td>
<td>27.80/0.8444</td>
<td>33.13/0.9416</td>
</tr>
<tr>
<td>A2F-M [37]</td>
<td>412K</td>
<td>33.99/0.9236</td>
<td>30.08/0.8367</td>
<td>28.91/0.8000</td>
<td>27.55/0.8385</td>
<td>-</td>
</tr>
<tr>
<td>AWSRN-S [35]</td>
<td>477K</td>
<td>34.02/0.9240</td>
<td>30.09/0.8376</td>
<td>28.92/0.8009</td>
<td>27.57/0.8391</td>
<td>32.82/0.9393</td>
</tr>
<tr>
<td>SRFBN-S [34]</td>
<td>483K</td>
<td>34.20/0.9255</td>
<td>30.10/0.8372</td>
<td>28.96/0.8010</td>
<td>27.66/0.8415</td>
<td>33.02/0.9404</td>
</tr>
<tr>
<td>IDN [41]</td>
<td>552K</td>
<td>34.11/0.9253</td>
<td>29.99/0.8354</td>
<td>28.95/0.8013</td>
<td>27.42/0.8359</td>
<td>-</td>
</tr>
<tr>
<td>VDSR [7]</td>
<td>665K</td>
<td>33.66/0.9213</td>
<td>29.77/0.8314</td>
<td>28.82/0.7976</td>
<td>27.14/0.8279</td>
<td>32.01/0.9310</td>
</tr>
<tr>
<td>MemNet [23]</td>
<td>677K</td>
<td>34.09/0.9248</td>
<td>30.00/0.8350</td>
<td>28.96/0.8001</td>
<td>27.56/0.8376</td>
<td>-</td>
</tr>
<tr>
<td>IMDN [64]</td>
<td>703K</td>
<td>34.36/0.9270</td>
<td>30.32/0.8417</td>
<td>29.09/0.8046</td>
<td>28.17/0.8519</td>
<td>33.61/0.9445</td>
</tr>
<tr>
<td>A2F-M [37]</td>
<td>1003K</td>
<td>34.50/0.9278</td>
<td>30.39/0.8427</td>
<td>29.11/0.8054</td>
<td>28.28/0.8546</td>
<td>33.66/0.9453</td>
</tr>
<tr>
<td>AWSRN-M [35]</td>
<td>1143K</td>
<td>34.42/0.9275</td>
<td>30.32/0.8419</td>
<td>29.13/0.8059</td>
<td>28.26/0.8545</td>
<td>33.64/0.9450</td>
</tr>
<tr>
<td>SelNet [42]</td>
<td>1159K</td>
<td>34.27/0.9257</td>
<td>30.30/0.8399</td>
<td>28.97/0.8025</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>A2F-L [37]</td>
<td>1367K</td>
<td>34.54/0.9283</td>
<td>30.41/0.8436</td>
<td>29.14/0.8062</td>
<td>28.40/0.8574</td>
<td>33.83/0.9463</td>
</tr>
<tr>
<td>AWSRN [35]</td>
<td>1476K</td>
<td>34.52/0.9281</td>
<td>30.38/0.8426</td>
<td>29.16/0.8069</td>
<td>28.42/0.8580</td>
<td>33.85/0.9463</td>
</tr>
<tr>
<td>SRMDNF [24]</td>
<td>1530K</td>
<td>34.12/0.9250</td>
<td>30.04/0.8370</td>
<td>28.97/0.8030</td>
<td>27.57/0.8400</td>
<td>-</td>
</tr>
<tr>
<td>CARN [32]</td>
<td>1592K</td>
<td>34.29/0.9255</td>
<td>30.29/0.8407</td>
<td>29.06/0.8034</td>
<td>28.06/0.8493</td>
<td>-</td>
</tr>
<tr>
<td>DRCN [8]</td>
<td>1774K</td>
<td>33.82/0.9226</td>
<td>29.76/0.8311</td>
<td>28.80/0.7963</td>
<td>27.15/0.8276</td>
<td>32.31/0.9328</td>
</tr>
<tr>
<td>MFCC (Ours)</td>
<td>1862K</td>
<td>34.67/0.9940</td>
<td>30.51/0.8456</td>
<td>29.22/0.8080</td>
<td>28.64/0.8616</td>
<td>34.15/0.9478</td>
</tr>
<tr>
<td>MSRN [33]</td>
<td>6114K</td>
<td>34.46/0.9278</td>
<td>30.41/0.8437</td>
<td>29.15/0.8064</td>
<td>28.33/0.8561</td>
<td>33.67/0.9456</td>
</tr>
</tbody>
</table>
For the most challenging upscale factor 8x, our model can continue as the best PSNR SR model, as shown in Table 6 and can improve a significant PSNR improvement of 0.91 dB with a comparable trade-off over the state-of-the-art model called EDSR.
Table 6 RESULTS QUANTITATIVE OF MODEL X8

<table>
<thead>
<tr>
<th>Model</th>
<th>Param</th>
<th>Set5</th>
<th>Set14</th>
<th>B100</th>
<th>Urban100</th>
<th>Manga109</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSRCNN [22]</td>
<td>12K</td>
<td>25.42/0.6440</td>
<td>25.34/0.5482</td>
<td>24.21/0.5112</td>
<td>21.32/0.5090</td>
<td>22.39/0.6357</td>
</tr>
<tr>
<td>SRCNN [21]</td>
<td>57K</td>
<td>25.34/0.6471</td>
<td>23.86/0.5443</td>
<td>24.14/0.5043</td>
<td>21.29/0.5133</td>
<td>22.46/0.6606</td>
</tr>
<tr>
<td>VDSR [7]</td>
<td>665K</td>
<td>25.73/0.6743</td>
<td>23.20/0.5110</td>
<td>24.34/0.5169</td>
<td>21.48/0.5229</td>
<td>22.73/0.6688</td>
</tr>
<tr>
<td>LapSRN [6]</td>
<td>813K</td>
<td>26.15/0.7028</td>
<td>24.45/0.5792</td>
<td>24.54/0.5293</td>
<td>21.81/0.5555</td>
<td>23.39/0.7068</td>
</tr>
<tr>
<td>DRCN [8]</td>
<td>1,774K</td>
<td>25.93/0.6740</td>
<td>24.25/0.5510</td>
<td>24.49/0.5170</td>
<td>21.71/0.5230</td>
<td>23.20/0.6690</td>
</tr>
<tr>
<td>AWSRN [35]</td>
<td>2,348K</td>
<td>26.97/0.7747</td>
<td>24.99/0.6414</td>
<td>24.80/0.5967</td>
<td>22.45/0.6174</td>
<td>24.60/0.7782</td>
</tr>
<tr>
<td>MFCC (Ours)</td>
<td>2,453K</td>
<td>27.07/0.7762</td>
<td>25.01/0.6412</td>
<td>24.84/0.5980</td>
<td>22.54/0.6196</td>
<td>24.63/0.7791</td>
</tr>
<tr>
<td>MSRN [33]</td>
<td>6,226K</td>
<td>26.59/0.7254</td>
<td>24.88/0.5961</td>
<td>24.70/0.5410</td>
<td>22.37/0.5977</td>
<td>24.28/0.7517</td>
</tr>
</tbody>
</table>

In addition, to the objective performance of our model, we reasonably describe the subjective quality of our model. As we can see in Figure 49, the visual quality of the SR image of our model is clear and sharp enough to follow the visual system of humans. As displayed in this figure, all models suffer over-smoothing degradation or cannot reconstruct the tiny lines. In contrast, our model outperforms compared to the other models. In addition, the highest PSNR and SSIM belong to our model.

In contrast to our result, other models produced blur results and showed weakness in reconstructing a sharp image. The PSNR and SSIM of our results are significantly better than other models. Figure 51 demonstrates the visual comparisons belonging to the Urban100 dataset. According to this figure, our MFCC result successfully generates the SR image similar to the GT image. In contrast, the other models’ results show weakness in reconstructing the lines identical to the reference image. The best PSNR and SSIM belong to our model.
<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR (dB)</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VDSR</td>
<td>18.25</td>
<td>0.4075</td>
</tr>
<tr>
<td>IMDN</td>
<td>20.31</td>
<td>0.5771</td>
</tr>
<tr>
<td>LapSRN</td>
<td>18.28</td>
<td>0.3715</td>
</tr>
<tr>
<td>AWSRN</td>
<td>28.18</td>
<td>0.9534</td>
</tr>
<tr>
<td>CARN</td>
<td>20.07</td>
<td>0.5682</td>
</tr>
<tr>
<td>MFCC (Our)</td>
<td>28.49</td>
<td>0.9460</td>
</tr>
<tr>
<td>MSRN</td>
<td>19.48</td>
<td>0.5223</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR (dB)</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VDSR</td>
<td>22.19</td>
<td>0.6181</td>
</tr>
<tr>
<td>IMDN</td>
<td>21.85</td>
<td>0.6083</td>
</tr>
<tr>
<td>LapSRN</td>
<td>22.59</td>
<td>0.6514</td>
</tr>
<tr>
<td>AWSRN</td>
<td>22.72</td>
<td>0.6891</td>
</tr>
<tr>
<td>CARN</td>
<td>22.52</td>
<td>0.6205</td>
</tr>
<tr>
<td>MFCC (Ours)</td>
<td>23.74</td>
<td>0.7801</td>
</tr>
<tr>
<td>MSRN</td>
<td>20.54</td>
<td>0.5194</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR (dB)</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VDSR</td>
<td>16.97</td>
<td>0.7630</td>
</tr>
<tr>
<td>IMDN</td>
<td>18.11</td>
<td>0.8101</td>
</tr>
<tr>
<td>LapSRN</td>
<td>18.49</td>
<td>0.8512</td>
</tr>
<tr>
<td>AWSRN</td>
<td>14.82</td>
<td>0.6038</td>
</tr>
<tr>
<td>CARN</td>
<td>18.39</td>
<td>0.8335</td>
</tr>
<tr>
<td>MFCC (Ours)</td>
<td>21.28</td>
<td>0.9093</td>
</tr>
<tr>
<td>MSRN</td>
<td>13.11</td>
<td>0.4398</td>
</tr>
</tbody>
</table>

Figure 49 Qualitative comparison over datasets for scale ×4. The red rectangle indicates the area of interest for zooming.
Figure 50 Qualitative comparison over datasets for scale $\times 4$. The red rectangle indicates the area of interest for zooming.
Figure 51 Qualitative comparison of img093 belongs to Urban100 dataset for scale x8. The yellow rectangle indicates the area of interest for zooming.

Figure 52 Qualitative comparison of an image belongs to Manga109 dataset for scale x8. The yellow rectangle indicates the area of interest for zooming.
Figure 52 compares the results of an image belonging to Manga109 datasets. The PSNR and SSIM of our proposed model are the highest. According to this figure, other models could not reconstruct the parallel lines located at the top of the selected patch and merge the lines. In contrast, our model shows a robust ability to produce tiny edges at this scale.

### 4.3. Result Analysis of Hardware

**Figure 53 Target Resolution**

Figure 53 shows three target resolution of the input LR image with 640x360 that the proposed hardware can support. They are x2, x3, and x4 scale factors with HD (1280x720), FHD (1920x1080), and QHD (2560x1440) FPS output size, respectively. The dataset for the proposed hardware can be found in Figure 45.

**Figure 54 Result of KV260 processing super resolution method MFCC 2x**
Figure 54 shows that the frame rate decreases from 45 fps to 30 fps due to the display process of the proposed hardware. Due to the utilizing computer vision tasks in a wide range of applications, single image super resolution is crucial in the computer vision task. Because of the constant increase in the network depth of super-resolution models recently, it has led to a massive amount of computation and memory utilization. On the other hand, the depth of these super-resolution models achieves a small increase in the results’ performance. To design a more effective SR model. The experimental results on four benchmark datasets demonstrates outperformance of our model in terms of speed and accuracy, compared to the other state-of-the-art methods. Besides, this model effectively decreasing the number parameters and computation time. Moreover, the progressive reconstruction approach improves result at higher scale and simultaneously provides a variety of scales in a single model. This model was evaluated on the benchmark image datasets and achieved efficient performance over the other state-of-the-art models.

Table 7 Hardware implementations average PSNR in with a scaling factor 2 of Set5

<table>
<thead>
<tr>
<th>Scale (FPS Output Size)</th>
<th>x2 (1280x720)</th>
<th>x3 (1920x1080)</th>
<th>x4 (2560x1440)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods</td>
<td>Param(k)</td>
<td>PSNR (dB)</td>
<td>FPS</td>
</tr>
<tr>
<td>RCAN (RG1, RCAB8) – Thread1</td>
<td>87</td>
<td>37.38</td>
<td>18</td>
</tr>
<tr>
<td>RCAN (RG1, RCAB8) – Thread2</td>
<td>19</td>
<td></td>
<td>19</td>
</tr>
<tr>
<td>MFCC (Our) – Thread1</td>
<td>20</td>
<td>37.20</td>
<td>39</td>
</tr>
<tr>
<td>MFCC (Our) – Thread2</td>
<td></td>
<td>45</td>
<td></td>
</tr>
</tbody>
</table>

Table 7 shows the number of parameters and average PSNR of the hardware implementation on scale 2, 3, and 8 of Set 5 for RCAN and MFCC in Thread1 and Thread2. The number of parameters for RCAN are 87, 98, and 96 while the number of parameters for MFCC are 20, 32, and 29 for x2, x3, and x4, respectively. Table 7 shows that MFCC consumes less parameters than RCAN about 77%, 67%, 70% for x2,
x3, and x4, respectively under a comparable PSNR. In addition, MFCC can achieve a higher frame rate than RCAN in Tread1 and Trade 2 for all scale factors.

Table 8 provides the important comparison of hardware implementation of the proposed algorithm compared with start-of-the-art algorithms. According to Table 8, it can be found that the supporting scale factors and number of target resolution of the proposed algorithm are more than others. Consequently, our algorithm is more suitable for several scenarios under a minimum hardware resource.

**Table 8 Hardware implementations comparison**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation</td>
<td>0.13 um CMOS</td>
<td>90 mm CMOS</td>
<td>Xilinx XCU1040</td>
<td>6.13 um CMOS Xilinx XCU1040</td>
<td>65 nm Xilinx XCU1040</td>
<td>65 nm Xilinx XCU1040</td>
<td>Xilinx XCVU9P</td>
</tr>
<tr>
<td>Memory Size (KB)</td>
<td>235</td>
<td>92</td>
<td>572</td>
<td>194</td>
<td>53</td>
<td>255</td>
<td></td>
</tr>
<tr>
<td>Max. Frequency (MHz)</td>
<td>431</td>
<td>124.4</td>
<td>150</td>
<td>220</td>
<td>200</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>Supported Scale</td>
<td>x2, x3</td>
<td>x2</td>
<td>x2, x4</td>
<td>x2</td>
<td>x2</td>
<td>x2, x3, x4</td>
<td></td>
</tr>
<tr>
<td>PSNR (dB)</td>
<td>34.00</td>
<td>-</td>
<td>36.66</td>
<td>36.49</td>
<td>37.50</td>
<td>37.50</td>
<td></td>
</tr>
<tr>
<td>HW Results</td>
<td>33.83</td>
<td>34.78</td>
<td>35.12</td>
<td>36.51</td>
<td>36.42</td>
<td>37.20</td>
<td></td>
</tr>
<tr>
<td>Target Resolution</td>
<td>4K UHD (30 fps)</td>
<td>4K UHD (60 fps)</td>
<td>4K UHD (30 fps)</td>
<td>4K UHD (60 fps)</td>
<td>4K UHD (30 fps)</td>
<td>4K UHD (60 fps)</td>
<td>HD, 1080p</td>
</tr>
</tbody>
</table>

*Set 5 x2*
Chapter 5
Conclusion and Future Works

This research proposed a lightweight single-image super-resolution model based on constructing Residual Group blocks on a multi-path residual architecture (MFCC). Utilizing a multi-path residual network increases the efficiency of the proposed lightweight model. Additionally, we addressed the lack of low-frequency details problem by employing the pixel shuffle fusion method. Based on this approach, the low-frequency details of the early layer are up-sampled and bypassed to the up-sampled features of the multi-path residual network. These layers' high and low-frequency information are fused and improve the proposed model's line and edge reconstructing capability. The experimental results on five benchmark datasets demonstrate our lightweight MFCC model outperforms other state-of-the-art models, especially on a scale of x8. For future work, the proposed model will be tuned to achieve the optimum parameters and then implemented on the FPGAs to support real-world applications.
REFERENCES


[10] H. Stark and P. Oskoui, "High-resolution image recovery from image-plane arrays,


NAME  Watchara Ruangsang
DATE OF BIRTH  12 March 1988
PLACE OF BIRTH  Bangkok
INSTITUTIONS ATTENDED  Chulalongkorn University (2015-2022)
Chulalongkorn University (2011-2015)
King Mongkut’s University of Technology North Bangkok (2008-2011)
HOME ADDRESS  14/5 M.4, Tha TalatSam, Phran District, Nakhon Pathom, 73210
Silver Medal Award from The International exhibition of Inventions Geneva on March 22, 2021.