



**AFRL-RY-WP-TR-2020-0268**

**SINGLE IMAGE SUPER RESOLUTION WITH INFRARED  
IMAGERY AND MULTI-STEP REINFORCEMENT  
LEARNING**

**Kyle Taylor Vassilo  
University of Dayton**

**SEPTEMBER 2020  
Final Report**

**Approved for public release; distribution is unlimited.**

*See additional restrictions described on inside pages*

**STINFO COPY**

**AIR FORCE RESEARCH LABORATORY  
SENSORS DIRECTORATE  
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7320  
AIR FORCE MATERIEL COMMAND  
UNITED STATES AIR FORCE**

<b>REPORT DOCUMENTATION PAGE</b>				<i>Form Approved</i> OMB No. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. <b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b></p>					
<b>1. REPORT DATE (DD-MM-YY)</b> September 2020		<b>2. REPORT TYPE</b> Thesis		<b>3. DATES COVERED (From - To)</b> 24 July 2017 –25 August 2020	
<b>4. TITLE AND SUBTITLE</b> SINGLE IMAGE SUPER RESOLUTION WITH INFRARED IMAGERY AND MULTI-STEP REINFORCEMENT LEARNING				<b>5a. CONTRACT NUMBER</b> FA8650-18-C-1073	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b> 61102F/63112F/63456F	
<b>6. AUTHOR(S)</b> Kyle Taylor Vassilo				<b>5d. PROJECT NUMBER</b> 3001/5327	
				<b>5e. TASK NUMBER</b> N/A	
				<b>5f. WORK UNIT NUMBER</b> Y1Q0	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b>  University of Dayton 300 College Park Dayton, OH 45469				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b>  Air Force Research Laboratory, Sensors Directorate Wright-Patterson Air Force Base, OH 45433-7320 Air Force Materiel Command United States Air Force				<b>10. SPONSORING/MONITORING AGENCY ACRONYM(S)</b> AFRL/Ryat	
				<b>11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)</b> AFRL-RY-WP-TR-2020-0268	
<b>12. DISTRIBUTION/AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited.					
<b>13. SUPPLEMENTARY NOTES</b> PAO case number 88ABW-2020-2665, Clearance Date 25 August 2020. © 2020 Kyle Taylor Vassilo. Presented in partial fulfillment of the requirements for the Degree of Master of Science in Electrical Engineering at The School of Engineering of the University of Dayton.. This work was funded in whole or in part by Department of the Air Force. The U.S. Government has for itself and others acting on its behalf an unlimited, paid-up, nonexclusive, irrevocable worldwide license to use, modify, reproduce, release, perform, display, or disclose the work by or on behalf of the U.S. Government. Report contains color.					
<b>14. ABSTRACT</b>  Recent studies have shown that Deep Learning (DL) algorithms can significantly improve Super Resolution (SR) performance. Single image SR is useful in producing High Resolution (HR) images from their Low Resolution (LR) counterparts. The motivation for SR is the potential to assist algorithms such as object detection, localization, and classification. Insufficient work has been conducted using Generative Adversarial Networks (GANs) for SR on infrared (IR) images despite its promising ability to increase object detection accuracy by extracting more precise features from a given image. This work adopts the idea of a relativistic GAN that utilizes Residual in Residual Dense blocks (RRDBs) for feature extraction, a novel residual image addition, and a Pixel Transposed Convolutional Layer (PixelTCL) for up-sampling. Recent work has validated the use of GANs for Visible Light (VL) images, making them a strong candidate. The inclusion of these components produce more realistic and natural features while also receiving superior metric values. Supplemental research applies a multi-agent Reinforcement Learning (RL) algorithm to Single Image Super-Resolution (SISR), creating an advanced ensemble approach for combining powerful GANs. In our implementation each agent chooses a particular action from a fixed action set comprised of results from existing GAN SISR algorithms to update its pixel values. The pixel-wise arrangement of agents and rewards encourages the algorithm to learn a strategy to increase the resolution of an image by choosing the best pixel values from each option.					
<b>15. SUBJECT TERMS</b> deep learning, reinforcement learning, SISR, generative adversarial networks (GAN), image processing, super resolution					
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT:</b> SAR	<b>18. NUMBER OF PAGES</b> 49	<b>19a. NAME OF RESPONSIBLE PERSON (Monitor)</b> Asif Mehmood <b>19b. TELEPHONE NUMBER (Include Area Code)</b> N/A
<b>a. REPORT</b> Unclassified	<b>b. ABSTRACT</b> Unclassified	<b>c. THIS PAGE</b> Unclassified			

SINGLE IMAGE SUPER RESOLUTION WITH INFRARED IMAGERY AND  
MULTI-STEP REINFORCEMENT LEARNING

Thesis  
Submitted to  
The School of Engineering of the  
UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for  
The Degree of  
Master of Science in Electrical Engineering

By  
Kyle Taylor Vassilo  
Dayton, Ohio  
August, 2020



SINGLE IMAGE SUPER RESOLUTION WITH INFRARED IMAGERY AND  
MULTI-STEP REINFORCEMENT LEARNING

Name: Vassilo, Kyle Taylor

APPROVED BY:

---

Tarek M. Taha, Ph.D.  
Advisory Committee Chairman  
Professor, Electrical and Computer  
Engineering

---

Asif Mehmood, Ph.D.  
Committee Member  
Research Scientist, Air Force  
Research Laboratory

---

Eric J. Balster, Ph.D.  
Committee Member  
Professor and Chair, Electrical and  
Computer Engineering

---

Robert J. Wilkens, Ph.D., P.E.  
Associate Dean for Research and Innovation  
Professor  
School of Engineering

---

Eddy M. Rojas, Ph.D., M.A., P.E.  
Dean, School of Engineering

© Copyright by  
Kyle Taylor Vassilo  
All rights reserved  
2020

# ABSTRACT

## SINGLE IMAGE SUPER RESOLUTION WITH INFRARED IMAGERY AND MULTI-STEP REINFORCEMENT LEARNING

Name: Vassilo, Kyle Taylor  
University of Dayton

Advisor: Dr. Tarek M. Taha

Recent studies have shown that Deep Learning (DL) algorithms can significantly improve Super Resolution (SR) performance. Single image SR is useful in producing High Resolution (HR) images from their Low Resolution (LR) counterparts. The motivation for SR is the potential to assist algorithms such as object detection, localization, and classification. Insufficient work has been conducted using Generative Adversarial Networks (GANs) for SR on infrared (IR) images despite its promising ability to increase object detection accuracy by extracting more precise features from a given image. This work adopts the idea of a relativistic GAN that utilizes Residual in Residual Dense blocks (RRDBs) for feature extraction, a novel residual image addition, and a Pixel Transposed Convolutional Layer (PixelTCL) for up-sampling. Recent work has validated the use of GANs for Visible Light (VL) images, making them a strong candidate. The inclusion of these components produce more realistic and natural features while also receiving superior metric values. Supplemental research applies a multi-agent Reinforcement Learning (RL) algorithm to Single Image Super-Resolution (SISR), creating an advanced ensemble approach for combining powerful GANs. In our implementation each agent chooses a particular action from a fixed action set comprised of results from existing GAN SISR algorithms to update its pixel values. The pixel-wise arrangement of agents and rewards encourages the algorithm to learn a strategy to increase the resolution of an image by choosing the best pixel values from each option.

## ACKNOWLEDGMENTS

I would like to convey my deepest appreciation for my advisory committee chairman, Dr. Tarek Taha. His continuous efforts to teach and lead created an environments for great success. Frequent meetings and positive advice helped me maintain a sharp focus on future goals that laid before me. Without your guidance, this thesis would have been entirely more difficult to complete.

I would also like to thank my additional committee members, Dr. Asif Mehmood and Dr. Eric Balster. Dr. Asif Mehmood is a member of the Air Force Research Laboratory, at Wright Patterson Air Force Base, who funded my research. I extend gratitude to Dr. Asif Mehmood for the opportunity to assist him in his research, while studying under his wealth of knowledge. I would like to acknowledge Dr. Eric Balster for his support in the Thesis process and establishing himself as an admirable leader.

A thank you to Dr. Patrick Hytla for supplying me with the appropriate supplies necessary for conducting my research and assisting in research ideas. A special thanks to Cory Heatwole whom I had the great pleasure of working alongside. During our time together, our work, *Multi-Step Reinforcement Learning for Single Image Super-Resolution*, was accepted by New Trends in Image Restoration and Enhancement (NTIRE) which is a Computer Vision and Pattern Recognition (CVPR) workshop that is co-sponsored by the Institute of Electrical and Electronics Engineers (IEEE). All rights under copyright were assigned to IEEE, but the authors are allowed to re-use the work.

I also thank my parents for the continual support through acquiring a higher degree. Their advice helped encourage me through the tough times and guided me through difficult decisions. Without their help this thesis would not have been possible.

# TABLE OF CONTENTS

ABSTRACT . . . . .	iii
ACKNOWLEDGMENTS . . . . .	iv
LIST OF FIGURES . . . . .	vi
LIST OF TABLES . . . . .	vii
CHAPTER I. INTRODUCTION . . . . .	1
CHAPTER II. INFRARED IMAGE SUPER RESOLUTION WITH GENERATIVE ADVERSARIAL NETWORKS . . . . .	3
2.1 Related Work . . . . .	5
2.2 Proposed Method . . . . .	7
2.2.1 Residual in Residual Dense Block . . . . .	9
2.2.2 Pixel Transposed Convolutional Layer . . . . .	10
2.2.3 Residual Image Addition . . . . .	12
2.2.4 Spectral Normalization . . . . .	13
2.2.5 Relativistic Average GAN . . . . .	13
2.3 Experiments . . . . .	14
2.3.1 Datasets . . . . .	14
2.3.2 Implementation Details . . . . .	15
2.3.3 Results . . . . .	16
2.4 Discussion . . . . .	20
CHAPTER III. MULTI-STEP REINFORCEMENT LEARNING FOR SINGLE IMAGE SUPER-RESOLUTION . . . . .	21
3.1 Related Work . . . . .	22
3.2 Proposed Method . . . . .	25
3.3 Experiments and Results . . . . .	29
3.3.1 VGG Rewards . . . . .	30
3.4 Discussion . . . . .	31
CHAPTER IV. CONCLUSION . . . . .	35
BIBLIOGRAPHY . . . . .	36



## LIST OF FIGURES

2.1	Generator Network . . . . .	8
2.2	Discriminator Network . . . . .	8
2.3	Residual in Residual Dense Block . . . . .	9
2.4	Dense Block . . . . .	9
2.5	Pixel Transposed Convolutional Layer . . . . .	11
2.6	Qualitative and Quantitative Test Results . . . . .	19
3.1	PixelRL Fully Convolutional Network Architecture . . . . .	24
3.2	Single Image Super-Resolution Action Set . . . . .	27
3.3	Super Resolution Action Map at Each Timestep . . . . .	27
3.4	Qualitative Comparison on Image img_005 from the Urban100 Dataset . . . . .	32
3.5	Qualitative Comparison on Image 0818 from the DIV2K Dataset . . . . .	33

## LIST OF TABLES

2.1	Quantitative Comparison on Set1 . . . . .	17
2.2	Quantitative Comparison on Set2 . . . . .	17
3.1	Quantitative Comparison in RGB Color Space . . . . .	30

# CHAPTER I

## INTRODUCTION

Since the creation of Convolutional Neural Networks (CNNs), they have been broadly used to develop solutions to many open ended problems. Scientists have now embraced CNNs as a powerful tool to temporarily solve these difficult problems. CNNs are most commonly used in image processing where they manipulate image data. Super Resolution (SR) is a sub-field within image processing that deals with enhancing the resolution of Low Resolution (LR) images. Before CNNs, each task required individual analysis and distinct parameters to receive acceptable results. For instance, when applying text recognition, multiple filters would have to be applied before sending the results to additional algorithms. With CNNs, one network combines these tasks into a single black box. Their goal is to model the input data with a complex function that manipulates the data and returns the desired results. SR concerns itself with attempting to reassemble High Resolution (HR) images from their LR counterparts. This work enables us to enhance the resolution of images with little high frequency information. High frequency images are sought after by many industries such as the medical, agricultural, and surveillance. These images can help experts locate and diagnose disease, locate worrisome vegetation, and help identify criminals. Chapter II analyzes the use of Generative Adversarial Networks (GANs) for SR on Infrared (IR) images. Most SR work deals with RGB images, which perceive settings similar to the human eye. However, IR images feature objects radiating thermal energy. The GAN combines two networks together, generator and discriminator, to fabricate intermediate points. The proposed network models the ESRGAN network to up-sample IR images [1]. A new Pixel Transposed Convolutional Layer (PixelTCL) is used for up-sampling which has yet to be used in a GAN setting, and a novel Residual Image Addition that adds the residual image back to the main branch of the network before up-sampling [2]. Compared

to competing algorithms, these components produce results with higher metric values that appear more accurate to the original image.

SR is such an open-ended problem, that numerous CNN architectures have been developed to solve the problem. Chapter II applies Reinforcement Learning (RL) to SR. The network models the pixelRL network to increase the resolution of an image by selecting a certain action for each pixel [3]. PixelRL is a multi-agent Reinforcement Learning (RL) algorithm where each pixel is governed by an independent agent, which choose individual actions. This network distinguishes itself from previous work by applying the multiple agent RL to SR. An initial step is created, which up-samples the image before applying RL. There is also a novel action set that contains multiple high quality GAN outputs. This design creates a HR image by compiling numerous GAN options. By using multiple GAN options, superior metric vales were obtained. The perceptual quality of the resulting images were also preferable to the images used in the action set.

## CHAPTER II

### INFRARED IMAGE SUPER RESOLUTION WITH GENERATIVE ADVERSARIAL NETWORKS

Super Resolution (SR) is an expanding problem that attempts to reassemble High Resolution (HR) images from their Low Resolution (LR) counterparts. In recent years, research has highly depended on Deep Learning (DL) algorithms to learn a set of parameters that map LR input images to HR output images. These HR output images are highly sought after by subject areas that range from medical imaging to security. Some disciplines require images that capture unique frequencies of electromagnetic radiation on the electromagnetic spectrum. For instance, medical imaging sensors might receive increased energy signals such as X-rays. However, Infrared (IR) images accept frequencies lower than the visible light spectrum. IR sensors succeed in detecting infrared radiation by analyzing signals with wavelengths longer than those found in the visible light spectrum. These sensors can highlight objects radiating thermal energy and therefore allow us to perceive items hidden to the human eye. While IR imaging excels in highlighting thermal energy it lacks in retaining high frequency information, making the SR problem more difficult. However, the main goal of IR SR is to increase the accuracy of additional algorithms (object detection, localization, and classification) by supplying HR images.

In the past, researchers have made monumental strides in the field of SR. Most of this research has been conducted using DL algorithms to provide a versatile technique, more efficient than previous methods. Generative Adversarial Networks (GANs) are one of the novel ideas, designed by Goodfellow in 2014 [4], that have changed the way researchers look at DL. They have proved themselves in the realm of visible light (VL) imagery, by mimicking distributions of image data [5]. We hope that their success with VL imagery provides a

hopeful approach for IR images. Therefore, we will apply a GAN to increase the resolution of IR images. GANs are composed of two opposing Convolutional Neural Networks (CNNs), Generator and Discriminator, that cooperate to fabricate data [6]. The generator consists of multiple residual blocks that extract features from the LR image, followed by 2 up-sampling blocks that increase the dimensionality of LR images. These network elements cooperate to fabricate additional data points by mimicking the distribution of the input image [5]. The discriminator consists of convolutional layers matched with batch normalization layers, ending with a sigmoid function. It differentiates the real images from the fake, where the real image is the target image and the fake image is the image generated by the generator from the down-sampled image. GANs benefit from the adversarial competition between networks [6], where the discriminator acts as a learnable loss function for the overall GAN. It evaluates the generator's performance, of counterfeiting images, and updates its weights accordingly [6].

The novel contributions of this work are as follows:

- Proposing to inherit components of the ESRGAN network to up-sample IR images while retaining high frequency and IR information [1].
- Proposing to use a Pixel Transposed Convolutional Layer to up-sample the generated images [2]. This technique has yet to be used in a GAN setting.
- Proposing a Residual Image Addition, where the residual image is calculated and added back to the main branch of the network before up-sampling, to ensure high frequency information propagates through the network.

## 2.1 Related Work

Xudong et al. combines Compressive Sensing (CS) techniques with CNNs to enhance IR images [7]. CS is a signal processing technique that keeps the most significant high frequency components from LR images while throwing away the least significant. The super-resolved image is then sent through a CNN to learn adequate weights to effectively remove unwanted noise and salvage lost high frequency samples [7]. This CNN uses residual blocks that are used to add certain features back into the super-resolved images, providing more accurate results with higher PSNR and SSIM values [7]. The up-sampled image is then combined with the output of the CNN to give a final super-resolved image [7]. Another approach uses a CNN to produce a more accurate image by fusing IR and VL images of a single setting. Since the images are taken simultaneously, the generated image incorporates certain details from the VL image while maintaining the IR heat signatures of the captured area. Liu et al. proposes an architecture with Siamese CNNs that calculate shared weights [8]. The source images are decomposed into a Laplacian pyramid, fused together, and reconstructed [8]. These fused images give visually appealing features but alter the target image. The fused image seems to be of a lighter complexion and contains different characteristics than the original IR image. Zhong et al. split each source image (LR IR image and LR VL image) into high and low frequency sub-images [9]. The low frequency sub-images are then fused, as are the high frequency sub-images. These fused sub-images are then merged to create a single fused image [9]. Aside from IR SR, Kuanget et al. introduces a fully CNN able to remove optical noise [10]. Their architecture consists of two CNNs, a denoising network and a discriminator network, like a GAN [10]. The denoising network learns to produce more realistic results, while the discriminator network predicts how realistic the super-resolved

images are [10]. This architecture is successful at removing noise from an IR image, but fails to up-sample the image. Therefore, GANs will be analyzed for SR on IR images.

GANs have drawn much attention since their discovery in 2014 [4]. They are highly complex as they attempt to mimic the distribution of data [5]. Ma et al. proposes a FusionGAN that requires multiple images of the same area. It starts by sending a concatenation of VL and IR images to the generator of a GAN [11]. The discriminator then distinguishes the generated images from the original VL images, updating the generator to capture more high frequencies from the VL images [11]. In most cases, this algorithm is inapplicable for SR procedures because of its dual image requirement. Axel-Christian Guei et al. presents a SRGAN devised to up-sample facial IR images, called DeepSIRF 2.0 [12]. Their network consists of a generator with eight residual blocks that contain convolutional layers with batch normalization, and 2 pixel shuffling blocks used for up-sampling [12]. The discriminator consists of seven convolutional layers matched with batch normalization layers, ending with a sigmoid function [12]. Wang et al. suggest an Enhanced Super Resolution GAN (ESRGAN), for VL images, that introduces a new Residual in Residual Dense Block (RRDB) [1]. This block is different than the original block as it encompasses multiple dense blocks, within a single RRDB, that are residually added back to the RRDB branch. The dense blocks consist of multiple convolutional layers with activation functions that are densely connected to each other [1]. These blocks work remarkably well by preserving features extracted from previous layers. In this article, Wang et al. outline a unique normalization function, known as Spectral Normalization (SN). SN was introduced by Takeru Miyato et al. to normalize layer weights inside the discriminator network using Lipschitz continuity to promote stability between the generator and discriminator networks [13]. The ESRGAN also adopts the idea of a Relativistic average GAN (RaGAN) from



Alexia Jolicoeur-Martineau [14]. This model’s main advantage is its ability to surpass standard SRGAN image quality in less iterations [14]. The ingenuity lies in the loss function of the generator and discriminator, where the probability that the real image is more realistic than the fake image is calculated [14].

The up-sampling procedure is another crucial element of SR algorithms. Multiple strategies are used to up-sample images including interpolation, convolutional transpose, and sub-pixel convolution. Interpolation is an image processing procedure that inserts intermediate pixels according to their neighboring values. Interpolation, paired with a subsequent convolutional layer, generates acceptable results when used within the generator of a GAN. Convolutional transpose is a technique that has been widely used in the field of SR, as it up-samples by functioning in the reverse manner of a conventional convolutional layer. This technique is notorious for leaving behind checkerboard artifacts. Sub-pixel convolution was proposed by Shi et al. to up-sample an image using a fractional stride to create additional data points [15]. Hongyang Gao et al. recently published a paper introducing a novel Pixel Transposed Convolutional Layer (PixelTCL) used to increase the dimensionality of LR images [2]. Its recognition comes from its proficiency of removing checkerboard artifacts from the up-sampled image by increasing intermediate relationships between convolutional layers. This technique has not been studied with respect to SR with GANs.

## 2.2 Proposed Method

The proposed model distinguishes itself from the original SRGAN with numerous components [6]. Its generator incorporates RRDBs [1], PixelTCLs [2], and a new residual image addition function. Examining the first convolutional layer of the generator, shown in Figure 2.1, we can see that the network uses a small kernel size and drops the activation

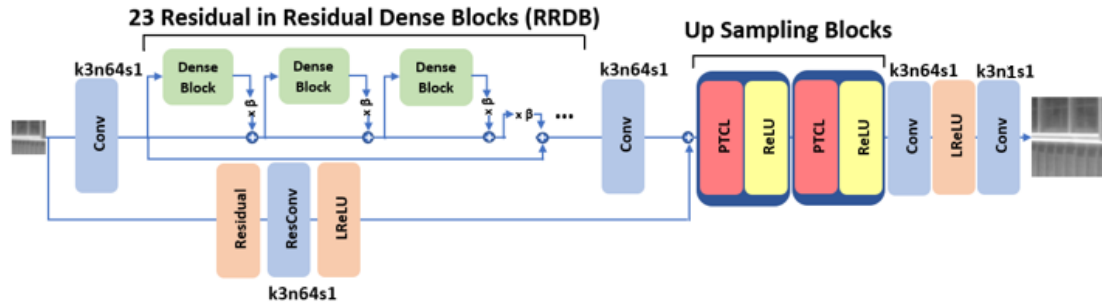


Figure 2.1: Network generator architecture with corresponding kernel size ( $k$ ), number of feature output maps ( $n$ ), and stride ( $s$ ) for each convolutional layer. A single RRDB is shown, but the generator contains 23 RRDBs that each include 3 dense blocks [1].

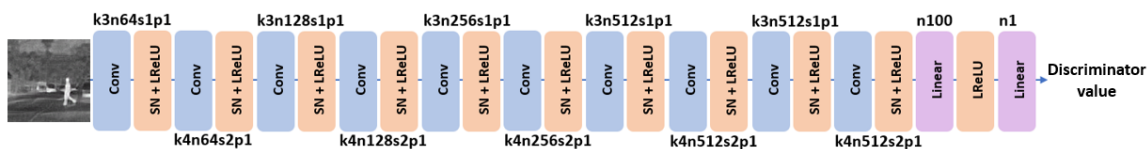


Figure 2.2: Network discriminator architecture with corresponding kernel size ( $k$ ), number of feature output maps ( $n$ ), stride ( $s$ ), and padding ( $p$ ) for each convolutional layer [1]. The discriminator follows a VGG orientation style.

function. This technique was copied from the ESRGAN, as it helps the network generate more accurate images with less iterations [1]. It was thought that a larger kernel size would help extract initial features from the image, but this turned out to be misleading. The three main pieces of the generator; the RRDBs, the PixelTCL, and the residual addition will be described in full detail.

The proposed model’s discriminator network adds two additional convolutional layers to mimic a VGG style architecture [1], utilizes SN [13], and removes the sigmoid function to introduce the relativistic average concept [14]. The discriminator’s design, illustrated by Figure 2.2, accepts single channel images and returns a single value. The network drops

### 2.3 Residual in Residual Dense Blocks (RRDB)

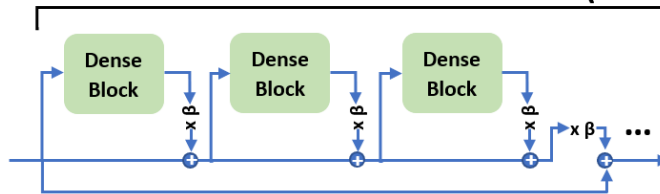


Figure 2.3: The RRDB used to extract features in the generator [1].

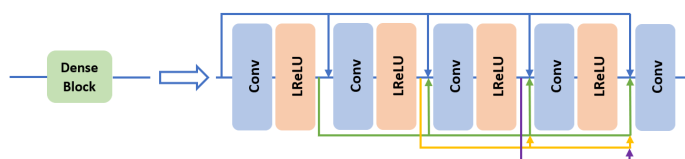


Figure 2.4: The architecture of a dense block within the RRDB [1].

the ending sigmoid function, enabling us to calculate the relativistic loss and replaces the batch normalization functions, of the original SRGAN, with SN for stability.

#### 2.2.1 Residual in Residual Dense Block

SRGANs use residual blocks to ensure certain features propagate through the system. It is vital to the success of the network to keep the extracted features in the super-resolved image. A single RRDB can be seen in Figure 2.3, which illustrates the residual in residual part of the RRDB. Each RRDB contains three residual dense blocks that are multiplied by a scalar value, in our case we chose 0.2 as the scalar value  $\beta$ , before being added back to the RRDB branch [1]. After the execution of each dense block, the RRDB branch is multiplied by the same constant  $\beta$  value and added back to the main branch, creating

stronger relationships between adjacent blocks. The components of a dense block can be seen in Figure 2.4. Each convolutional layer within the dense block uses a kernel size of 3, a stride of 1, and outputs 32 feature maps. The input number of feature maps grows as more and more previously calculated layers are concatenated to the input of subsequent layers. The functionality of the dense block can be duplicated using the following pseudocode:

$$\begin{aligned} Conv1 &= \text{LeakyReLU}(Conv(input)) \\ Conv2 &= \text{LeakyReLU}(Conv(input \oplus Conv1)) \\ Conv3 &= \text{LeakyReLU}(Conv(input \oplus Conv1 \oplus Conv2)) \\ Conv4 &= \text{LeakyReLU}(Conv(input \oplus Conv1 \oplus Conv2 \oplus Conv3)) \\ Conv5 &= \text{LeakyReLU}(Conv(input \oplus Conv1 \oplus Conv2 \oplus Conv3 \oplus Conv4)) \end{aligned}$$

where  $\oplus$  illustrates a concatenation operation. The dense connections characterized by this block are inherited from a GAN model introduced by Marc Bosch et al. [16]. These dense connections perform exceptionally well by maintaining a strong gradient flow within the network [16]. Figure 2.4 also reveals the absence of a normalization function. According to Yungang Zhang et al., removing the batch normalization function within residual blocks can improve its performance by expanding the model size [17], whereas Xintao Wang et al. explain how multiplying each block by a constant  $\beta$  value can avoid the instability of removing the batch normalization layers [1].

## 2.2.2 Pixel Transposed Convolutional Layer

Networks that require up-sampling data have recently been using techniques such as transposed convolution or sub-pixel convolution. Transposed convolution operates in the reverse manner of a normal convolutional layer. It creates a one-to-many relationship between the low resolution data and the high resolution data respectively. One value in the

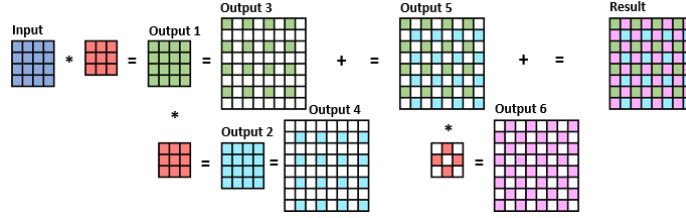


Figure 2.5: The PixelTCL used to up-sample image data in the generator [2].

input tensor is mapped to many values in the output tensor by multiplying its value by all existing values in the kernel and is laid on top of the output tensor. If the stride is small enough that the kernels overlap, multiple values are added to obtain the final value for that corresponding index of the output tensor.

Instead of performing convolution on the up-sampled image, sub-pixel convolution performs its convolution on the LR image to decrease the amount of computational parameters. Shi et al. performed convolution on the LR image to obtain several feature maps that were then shuffled to create an up-sampled image [15]. This increases the image dimensionality by populating an area on the super-resolved image with the same pixel of each feature map [15]. Both techniques have the potential of adding artifacts to the super-resolved image, representing a checkerboard pattern; therefore, the proposed network integrates 2 PixelTCLs that Hongyang Gao et al. guarantee will eliminate this pattern [2]. Figure 2.5 demonstrates the PixelTCL up-sampling method as it begins by performing a convolution with a kernel size of 3. The resulting tensor is then dilated by inserting rows and columns of zeros to construct a tensor double the size. This dilated tensor is illustrated by output3 in Figure 2.5. A second convolution is performed on output1 to produce output2. This tensor is then dilated and shifted to provide output4. The dilated tensors are then added

together to construct a tensor where every other value is zero. The next convolution exploits a masked kernel where certain values are set to 0. This convolution generates the missing values which are then added to output5 to create the desired output. This technique ensures that each pixel in the resulting image has a strong relationship with its adjacent neighbors; therefore, excluding the checkerboard pattern. The results will prove beneficial as this is a novel technique with respect to SR with GANs.

### 2.2.3 Residual Image Addition

In image processing, a residual image preserves the high frequency components of an image by subtracting a sampled image from the original image. In SR, the high frequency elements are what the generated image is missing. To ensure that the high frequency components are not dropped, the residual image is added back to the main branch of the network before up-sampling, similar to Yewen Sun et al. network [18]. The residual image is created by first down-sampling the input image (plus bicubic interpolation) from a  $32 \times 32$  sized image to a  $24 \times 24$  sized image, and re up-sampling it using bicubic interpolation. This image is then subtracted from the original LR input image to obtain the residual image. The residual image is then fed through a convolutional layer to extract high frequency features, from the residual image, before being added back to the main branch of the network. This approach differentiates itself from Yewen Sun et al. network by using a different sampling ratio, a shallower skip connection, and adding the results back to the main branch of the network before up-sampling.

## 2.2.4 Spectral Normalization

The original SRGAN uses batch normalization to normalize the data into values of the same range [6]. Batch Normalization helps a network train faster and go deeper by feeding similar distributions of data to the activation functions that follow. This decreases the amount of training time by creating similar distributions for each layer. SN is a technique used to recalculate the weight of a layer ( $W$ ) [13]. These normalized weights help stabilize the discriminator of a GAN by dividing each weight value by  $\sigma(W)$ :

$$\sigma(W) = \sqrt{\lambda} \tag{2.1}$$

where  $\lambda$  is the largest eigenvalue of  $W^T W$  [13]. This eigenvalue can be found using the following:

$$(W^T W)\bar{x} = \lambda\bar{x} \tag{2.2}$$

where  $\bar{x}$  is the eigenvector and  $\lambda$  is the corresponding eigenvalue.

## 2.2.5 Relativistic Average GAN

The original SRGAN employs a discriminator that distinguishes real from fake images. It calculates the probability that the given data is real (returns a 1 if it believes it is an original image from the dataset). The SRGAN losses are measured by:

$$L_D^{SGAN} = -E_{x_r \sim P}[\log(\text{sigmoid}(C(x_r)))] - E_{x_f \sim Q}[\log(1 - \text{sigmoid}(C(x_f)))] \tag{2.3}$$

$$L_G^{SGAN} = -E_{x_f \sim Q}[\log(\text{sigmoid}(C(x_f)))] \quad (2.4)$$

where  $L_D^{SGAN}$  is the discriminator's loss and  $L_G^{SGAN}$  is the generator's loss [14]. In equation 2.3 and equation 2.4,  $x_r$  and  $x_f$  are the reference image and generated image, respectively. The  $C(\cdot)$  function represents the discriminator without its final activation function. The relativistic GAN's discriminator manipulates both loss functions by measuring the distance between the real and fake data [14]. This is done by subtracting the real and the fake data received from the discriminator. Going a step further, the RaGAN calculates the probability that the real data is more realistic than the fake data, just like the relativistic GAN, but the average of the opposite label is then subtracted from the relativistic output [14]. The RaGAN is calculated by:

$$\begin{aligned} L_D^{RaGAN} = & - E_{x_r \sim P}[\log(\text{sigmoid}(C(x_r) - E_{x_f \sim Q}C(x_f)))] \\ & - E_{x_f \sim Q}[\log(1 - \text{sigmoid}(C(x_f) - E_{x_r \sim P}C(x_r)))] \end{aligned} \quad (2.5)$$

$$\begin{aligned} L_G^{RaGAN} = & - E_{x_f \sim Q}[\log(\text{sigmoid}(C(x_f) - E_{x_r \sim P}C(x_r)))] \\ & - E_{x_r \sim P}[\log(1 - \text{sigmoid}(C(x_r) - E_{x_f \sim Q}C(x_f)))] \end{aligned} \quad (2.6)$$

which illustrates the distance from real to fake data [14]. The subtraction of the average is demonstrated inside the sigmoid function of equation 2.5 and equation 2.6.

## 2.3 Experiments

### 2.3.1 Datasets

The IR training images are taken from the FLIR dataset. These images are randomly cropped using MATLAB to produce 20,000 128×128 training images. These images are



initially sent through a Gaussian filter with a standard deviation of  $\sigma = 1.3$ , down-sampled by a factor of 4, and interpolated using bicubic interpolation [19]. The Gaussian filter is used to represent the image being taken from a camera at a further distance. The images are then concatenated to produce a three-channel image, where each channel contains the same values. This is required to calculate the perceptual loss (VGG loss). Therefore, the generator accepts a 3 channel image and outputs a single channel image.

The IR testing dataset hand picks 1,200 images from the FLIR testing dataset that are randomly cropped twice to create two separate testing datasets. Following the cropping, images with little to no high frequency content are thrown out, leaving us with two testing datasets, Set 1 containing 312 images and Set 2 containing 211 images.

### 2.3.2 Implementation Details

The network was run on 2 NVIDIA RTX 2080ti GPUs for 50 epochs (62,500 iterations) with a batch size of 16, using the stochastic optimization method known as Adam with decay rates of 0.9. The learning rate for both the generator and discriminator start at  $2 \times 10^{-4}$  while being reduced by a factor of 2 every 10 epochs (12,500 iterations). The generator is pre-trained for 2 epochs using Mean Squared Error (MSE). After the 2 initial epochs, the generator weights are saved and loaded back into the designed model where the Binary Cross Entropy (BCE) function is used. BCE is measured by:

$$BCE = -\frac{1}{N} \sum_{i=1}^N (y_{label} \times \log(D(y_{input})) + (1 - y_{label}) \times \log(1 - D(y_{input}))) \quad (2.7)$$

where  $y_{label}$  is the target value and  $y_{input}$  is the input image. The generator and discriminator loss functions integrate the relativistic idea with BCE by computing the relativistic

operations before sending it to the BCE function. The algorithm utilizes the BCEWithLogitsLoss function, from PyTorch, that combines a sigmoid function with the BCE loss function that reconstructs more accurate images than the MSE loss function. The network's generator loss function consists of BCE loss, per-pixel loss, and perceptual loss [1]. The perceptual loss is calculated using a pre-trained VGG19 network and measuring the Least Squares Error (L2) between the real image feature space and the generated image feature space. The per-pixel loss is calculated by measuring L2 distance between the real image and the generated image. L2 is used because it seems to generate higher metric values than Least Absolute Deviations (L1).

### 2.3.3 Results

The results will be compared to the DeepSIRF 2.0 network introduced by Axel-Christian Guei et al. [12], bicubic interpolation, nearest neighbor interpolation, and an ablation network. The 'Proposed Ablation Network' removes the pre-training procedure, replaces PixelTCL with Pixel Shuffle, and removes the residual image addition technique in the global skip connection to show how much these techniques add to the overall performance. The quantitative results are evaluated using PSNR, SSIM, MSE, and PIQE. PIQE is a fairly new metric provided through MATLAB. It is a no-reference quality metric that returns high values for images that are blurred or contain large amounts of noise and therefore a lower score is more acceptable; under 20 being exceptionally well. MSE and PSNR have been used extensively to measure image restoration. MSE measures the distances between each pixel value while PSNR builds from MSE by dividing the maximum pixel value by the reconstructed images' MSE value. SSIM was designed to measure a perceived change to an image. While MSE and PSNR measure the absolute difference of an image, SSIM tends to

measure the structural degradation of an image. The average quantitative results over both testing sets are presented in Table 2.1 and Table 2.2.

Table 2.1: Quantitative comparison between the proposed method and other competing SR algorithms on set1 (Best results in **bold**, second best in *italics*).

Up-Sampling Technique	PSNR	SSIM	MSE	PIQE
High Resolution	$\infty$	1.00	0.00	36.99
Nearest Neighbor	25.32	0.605	218.93	65.21
Bicubic	26.25	0.654	179.47	92.35
DeepSIRF 2.0	<i>26.48</i>	<i>0.662</i>	<i>170.06</i>	78.81
Proposed Ablation Network	<i>26.48</i>	0.661	173.47	<b>26.94</b>
Proposed Network	<b>26.94</b>	<b>0.687</b>	<b>156.71</b>	<i>31.54</i>

Table 2.2: Quantitative comparison between the proposed method and other competing SR algorithms on set2 (Best results in **bold**, second best in *italics*).

Up-Sampling Technique	PSNR	SSIM	MSE	PIQE
High Resolution	$\infty$	1.00	0.00	40.47
Nearest Neighbor	25.05	0.600	228.87	65.12
Bicubic	25.96	0.651	188.22	91.39
DeepSIRF 2.0	<i>26.24</i>	<i>0.661</i>	<i>177.24</i>	80.03
Proposed Ablation Network	26.15	0.660	184.20	<b>28.57</b>
Proposed Network	<b>26.66</b>	<b>0.685</b>	<b>165.49</b>	<i>34.03</i>

A benchmark testing dataset is desperately needed for IR SR. This benchmark will allow easy comparison between methods, without having to recreate competing methods separately. As of now there is no such IR testing benchmark, so a custom testing dataset was created for comparison. The proposed method achieved the best quantitative scores in Table 2.1, for PSNR, SSIM, and MSE. Subsequently, the proposed model also performed the

best in Table 2.2, being that the two sets come from the same set of images. The ablation network scores the best in PIQE, with the proposed network close behind. Looking at Figure 2.6, it is easy to see that these two methods create the most realistic results. Therefore, we propose to use PIQE as the main metric for IR SR. The ablation network and proposed network results look similar in Figure 2.6, but at a closer look we can see that the proposed network is cleaner around edges and other high frequency areas. In Table 2.1 and Table 2.2, the DeepSIRF 2.0 network scores better than the ablation network in PSNR, SSIM, and MSE. However, the images created by the DeepSIRF 2.0 network look notably worse because of its inability to mimic the data distribution of the training set. These images illustrate a blurry characteristic that contributes to their poor PIQE scores. This is another reason why PIQE should have a larger influence on IR SR. Bicubic interpolation also scored well in every category except PIQE. The images generated by bicubic interpolation have lost their high frequency components. Therefore, it receives good PSNR, SSIM, and MSE scores because the low frequency components are very similar to that of the original image, but yields a poor PIQE score due to the generated images blurred appearance. The nearest neighbor interpolation performs exactly how we would expect for PSNR, SSIM, and MSE. These results are fairly low due to the images' discontinuity. The smooth contrast found in the HR image is entirely ruined by this interpolation technique. Patches of a single intensity are introduced where the smooth contrast use to be. Surprisingly, its PIQE score is better than bicubic interpolation and the DeepSIRF 2.0 network. PIQE scores blurred and noisy images poorly, but the blurred images receive a worse score. It is obvious to human inspection that nearest neighbor interpolation is inadequate to up-sample images, so it is easy to throw out. However, when it comes to high level representation methods, PIQE becomes an impressive metric to score the results.











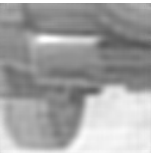




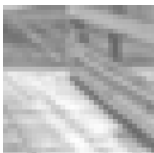






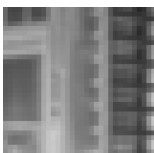





Original Image	Nearest Neighbor	Bicubic	DeepSIRF 2.0	Proposed Ablation Network	Proposed Network	HR
						
PSNR	27.37	28.71	28.95	<i>30.56</i>	<b>30.91</b>	$\infty$
SSIM	0.704	0.751	0.747	<i>0.801</i>	<b>0.811</b>	1.00
MSE	119.16	87.58	82.90	<i>57.20</i>	<b>52.72</b>	0.00
PIQE	69.70	92.98	77.02	<b>42.80</b>	<i>43.17</i>	48.04
						
PSNR	29.39	31.82	32.42	<i>33.52</i>	<b>33.88</b>	$\infty$
SSIM	0.775	0.846	0.835	<i>0.877</i>	<b>0.886</b>	1.00
MSE	74.78	42.77	37.25	<i>28.91</i>	<b>26.62</b>	0.00
PIQE	68.82	92.51	77.97	<i>41.24</i>	<b>40.22</b>	61.99
						
PSNR	24.26	25.19	25.08	<i>25.97</i>	<b>26.20</b>	$\infty$
SSIM	0.536	0.599	0.593	<i>0.662</i>	<b>0.677</b>	1.00
MSE	243.68	196.71	202.04	<i>164.42</i>	<b>156.07</b>	0.00
PIQE	48.51	90.06	79.77	<i>24.45</i>	<b>23.58</b>	34.14
						
PSNR	26.13	27.64	28.31	<i>28.56</i>	<b>29.56</b>	$\infty$
SSIM	0.681	0.769	<i>0.791</i>	0.789	<b>0.816</b>	1.00
MSE	158.47	112.04	95.95	<i>90.67</i>	<b>72.02</b>	0.00
PIQE	66.79	93.71	80.82	<b>46.17</b>	<i>65.47</i>	52.91

Figure 2.6: Qualitative and Quantitative test results (Best results in **bold**, second best in *italics*).

Figure 2.6 shows that the proposed method generates images of the highest quality, backed by their metric values. The proposed model preserves most of the textural characteristics, of the LR image, that have not been made entirely obsolete by down-sampling. The details in certain images have been recreated, allowing us to distinguish objects in the super-resolved image.

## 2.4 Discussion

The proposed model is based off the SRGAN algorithm consisting of two convolutional neural networks [6]. The generator’s residual network adopts its architecture from the ESRGAN [1] and its up-sampling procedure from Hongyang Gao et al. [15]. These components help the network produce images of higher quality by creating more connections between layers that aid in learning to mimic the data’s distribution. The residual image addition and pre-training techniques also add to the generator’s success. Comparing the proposed network to the ablation network, we can see that these features help the network learn a better representation. The relativistic idea increases the GAN’s efficiency by stabilizing the adversarial competition between networks within a GAN. The proposed method produces results that achieve higher metric values while also appearing more accurate to the original image. Most of the high frequency components from the original image are present in the generated image, even though IR images lack certain high frequency information. The proposed method is very powerful and can be applied to IR image SR to generate highly accurate images that retain IR information.

CHAPTER III  
MULTI-STEP REINFORCEMENT LEARNING FOR SINGLE IMAGE  
SUPER-RESOLUTION

Single Image Super-Resolution (SISR) is a vague problem that poses challenges in many computer vision applications. In this expanding problem we attempt to reproduce High-Resolution (HR) images from their Low-Resolution (LR) counterparts. In past years, SISR research revolved around Deep Learning (DL) algorithms that attempted to directly map LR input images to HR output images with a single pass through the network. Hundreds of unique Generative Adversarial Networks (GANs) have been applied to SISR that differ in their architectures, loss functions, and up-sampling techniques. They all share underlying fundamentals such as adversarial learning, Convolutional Neural Networks (CNNs), and attempting to mimic a certain distribution.

Instead of mapping LR images to their HR counterparts or imitating a distribution, the proposed method adopts a strategy to increase the resolution of an image by selecting a certain action for each pixel, based on pixelRL [3]. PixelRL is a multi-agent Reinforcement Learning (RL) algorithm where each pixel is modified by an independent agent. Agents choose particular actions to update their own pixel values. Being a RL technique, pixelRL permits the ability to analyze the output of the network through multiple iterations, referred to as timesteps. At each timestep the agent can choose to change its decision or stay the same. Furuta et al. applied this algorithm to three image processing tasks: denoising, restoration, and local color enhancement [3]. In our research we explore the possibility of applying and modifying pixelRL for the SISR problem.

The contributions of this work are as follows:

- Proposing a SISR RL network, which employs multiple agents. The network is given an up-sampled image, from a bicubic interpolation, and learns a policy for each pixel in order to maximize the total reward.
- Proposing a novel action set consisting of a number of GAN outputs, essentially creating an advanced ensemble approach to SISR.

Other SISR works, pixelRL, and some of its uses are described in Section 3.1. The proposed method is described in Section 3.2 and applications and results of the method are presented in Section 3.3. Finally, the paper is concluded in Section 3.4.

### 3.1 Related Work

GANs were first introduced by Ian Goodfellow et al., in 2014, and were applied to small grayscale and color images [4]. Consequently, GANs have become an extremely hot topic in DL research due to their impressive results and wide application range including: image editing [20] and texture removal (such as rain [21]) , stenographic security [22], data generation for other DL applications [23], attention prediction [24], and more. In 2016, Christian Ledig et al. proposed the SRGAN algorithm where they applied GANs to SISR [6]. Their generator consisted of 16 fully-convolutional residual blocks, ending with two PixelShufflers to increase the image dimension by a factor of 4. Their discriminator network consisted of 8 fully-convolutional blocks with batch normalization (in blocks 2-8) and LeakyReLU activations, followed by dense layers to produce the final real/fake prediction. To create more perceptually plausible images, the standard mean squared error (MSE) loss was replaced with a VGG loss, based on feature map activations from particular layers of the 19-layer VGG network, and an adversarial loss. Although they did not achieve



state-of-the-art PSNR or SSIM results, they set a new standard for photo-realistic quality by achieving the highest mean opinion score (MOS).

In the last few years, a number of image processing problems have been tackled with deep RL. Zhuopeng Li and Xiaoyan Zhang developed an image cropping algorithm which utilized collaborative deep RL [25]. Optimal cropping actions were chosen based on information from two different agents. Both agents were given a vector of all previous actions and an image. The first agent’s image was the current crop window of the original image while the second agent’s image was the current crop window of the emotional attention map associated with the original image. On two different datasets (FCD and HCD), Li and Zhang achieved state-of-the-art results. QingXing Cao et al. used deep RL for face hallucination, a specific case of SISR that applies only to face images [26]. A single agent determined the best patch of an image to enhance, via a custom CNN. The enhanced patch was then inserted back into the LR image and the agent selected a new patch. The process repeated until the entire image was enhanced. Their results showed that the order of patch enhancement affected the quality of the final image.

Ryosuke Furuta et al. used a novel multi-agent deep RL algorithm for image denoising, restoration, and color enhancement in which each pixel had its own agent, called pixelRL [3]. The action set for the denoising and restoration tasks included a number of standard image processing filters, pixel value increment and decrement, and “do nothing”. For color enhancement the action set included adjusting contrast, brightness, saturation, color balance, and “do nothing”. All tasks used the same CNN architecture, shown in Figure 3.1. Rewards were given to agents based on the MSE difference between the current image (state) and the target image and the previous image (state) and the target image (analogous to the  $L_2$  loss). Their denoising results (PSNR) were state-of-the-art for salt-and-pepper

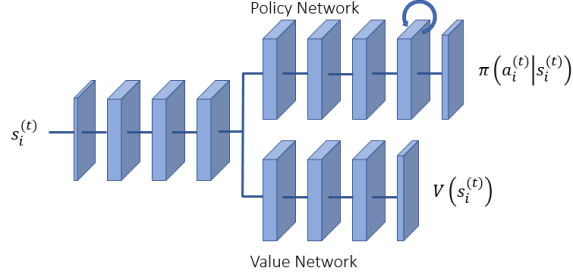


Figure 3.1: pixelRL FCN architecture [3].

noise with noise densities of 0.5 and 0.9. Their restoration results (PSNR and SSIM) were also state-of-the-art.

As of this writing, pixelRL has been adapted to two additional problem settings: MRI reconstruction [27] and 3D medical image segmentation [28]. According to Wentian Li et al., the key benefit of applying pixelRL to MRI reconstruction is its interpretability. Typical DL methods create a complex mapping from corrupted images to reconstructed images, making it nearly impossible for people to understand exactly *how* the image is reconstructed. In contrast, pixelRL uses a well-defined action set, allowing people to see exactly how each pixel of the image has been modified from corrupted to reconstructed. Similar to the original pixelRL, Li et al. used standard filters in their action set, along with “do nothing” and a pixel value decrement. They included a number of sharpening filters (Laplace, Sobel, and unsharp mask) with learnable continuous parameters. The original pixelRL network architecture, shown in Figure 3.1, was modified to accommodate these parameters by adding a third branch to the network after the split. Rewards were given based on the absolute error between the original image (state) and the target image and the final image (state) and the target image (analogous to the  $L_1$  loss). This method achieved

state-of-the-art results for NMSE and SSIM on the fastMRI dataset with a random 40% mask [27].

Xuan Liao et al. adapted pixelRL to 3D medical image segmentation by using voxel-wise agents and rewards. They also included user interaction to aid in the segmentation process, where users provide hints (such as points or bounding boxes) to the model. The network architecture used was nearly identical to that in pixelRL, shown in Figure 3.1, with the main difference being that the input to the final layer in the policy and value branches was a concatenation of all previous layers' outputs. Inputs to the network included a 3D image, the previous segmentation probabilities, and a user hint map. The new segmentation probabilities were created by tweaking the previous ones. As such, the action set consisted of various values that agents could use to modify the previous probabilities. With an output consisting of probabilities, a cross-entropy gain-based function was used to calculate the rewards based on the improvement from the previous output to the new output. This method achieved state-of-the-art results across three different datasets (BRATS2015, MM-WHS, AND NCI-ICBI2013) [28].

### 3.2 Proposed Method

PixelRL utilizes the asynchronous advantage actor-critic (A3C) algorithm [29]. A3C is a deep RL algorithm that uses multiple network instances that each have their own parameters and copy of the environment. This allows each instance to train in parallel and contribute to the shared learning of the global network. Actor-critic methods utilize two networks, a policy network and a value network, that simultaneously operate on the current state,  $s^{(t)}$ . The value network, known as the critic, outputs the expected total rewards,  $V(s^{(t)})$ , from the current state, which exhibits the quality of the current state [3].

The policy network, known as the actor, calculates the probabilities of the agent choosing each action,  $\pi(a^{(t)}|s^{(t)})$ , when in the current state [3].

PixelRL employs a Fully Convolutional Network (FCN) which rearranges the agents into a 2D space where they can share parameters [3]. This 2D representation facilitates the reward map convolution learning method proposed by Furuta et al [3]. In this method, the receptive field of an agent is treated as a weighted convolutional filter that influences the policies and values of neighboring agents. Figure 3.1 illustrates the pixelRL network architecture which applies a Gated Recurrent Unit (GRU) to the penultimate layer of the policy network. Cho et al. introduced the GRU in 2014 to implement a layer with memory that does not suffer from the vanishing gradient problem [30]. The last layer of the policy network produces a feature map for each action. A softmax function is applied across each of the feature maps to provide a probability distribution over the potential actions for each agent [3]. During training, the distribution is stochastically sampled to extract a single action for each pixel, while during testing the network always chooses the most probable action.

The pixelRL policy differs from traditional RL policies in that each pixel has an agent. Thus, the policy becomes  $\pi_i(a_i^{(t)}|s_i^{(t)})$ , where  $a_i^{(t)}$  and  $s_i^{(t)}$  are the action and state at timestep  $t$  of the  $i$ -th agent [3]. The number of actions come from the action set  $A$  predefined by the authors. In our case the action set consists of pixel value increment and decrement, “do nothing”, and choosing pixel values from images up-sampled by numerous GAN SISR algorithms. In the current implementation, the increment and decrement actions are applied to all channels of a pixel. PixelRL applied the filters during execution, however they could have used pre-filtered images, which inspired the proposed model’s actions. This idea is a novel contribution as we change the action set to a number of pre-computed GAN outputs.

	Action	Color
1	pixel value -= 1	Blue
2	do nothing	Cyan
3	pixel value += 1	Green
4	edsr baseline	Light Green
5	esrgan	Brown
6	esrgan-pnsr	Pink
7	ppon	Purple

Figure 3.2: SISR action set: 1. Subtract 1 from the value of all channels of the pixel, 2. Do not change the pixel value, 3. Add 1 to the value of all channels of the pixel, 4. Substitute pixel value with that of the Enhanced Deep Super-Resolution network (EDSR), 5. Substitute pixel value with that of the Enhanced Super-Resolution GAN (ESRGAN), 6. Substitute pixel value with that of the Enhanced Super-Resolution GAN, PSNR focused (ESRGAN-PSNR), 7. Substitute pixel value with that of the Progressive Perception-Oriented Network (PPON).

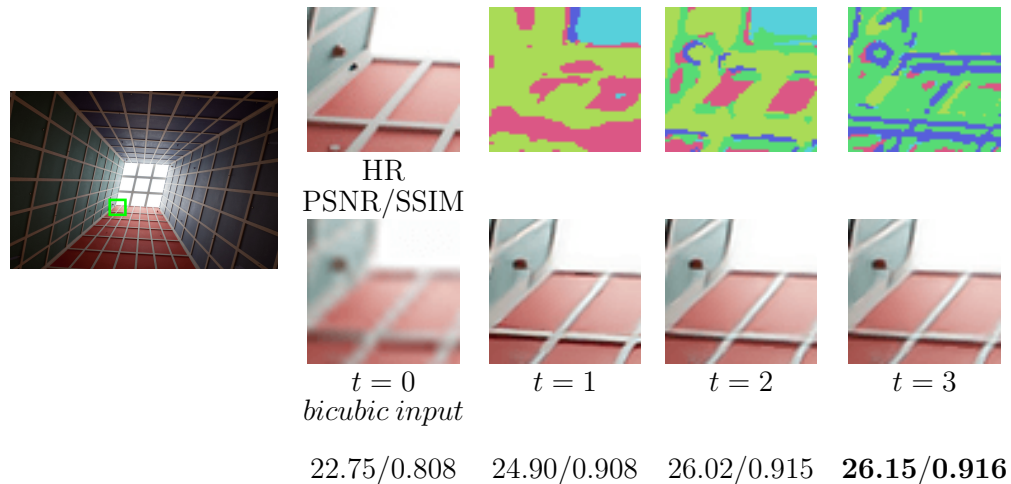


Figure 3.3: Super Resolution process and action map at each timestep for Urban100 image img\_090 (RGB color space, pixel-wise agents, MSE-based rewards). The colors represent the agent’s choice of options from Figure 3.2, essentially replacing the original image pixel with a different option.

These actions are illustrated in Figure 3.2, where the colors will be used to create action maps for each timestep in subsequent figures (see Figure 3.3 for an example). Selecting the GAN algorithms began with a survey of current SISR methods. Their results were

compared and ranked according to PSNR and SSIM values. The GANs selected for the proposed model’s action set [31, 32, 1] had the highest values amongst all of the algorithms with publicly available code.

Here the agent transitions from state to state by choosing an action and acquiring a reward to assist in learning an efficient policy,  $\pi = [\pi_1, \dots, \pi_N]$ , where  $N$  is the total number of pixels/agents in an image [3]. The reward function compares the output and previous images at each timestep with the target image, described as:

$$r_i^{(t)} = (I_i^{target} - y_i^{(t-1)})^2 - (I_i^{target} - y_i^{(t)})^2, \quad (3.1)$$

where  $r_i^{(t)}$  is the reward for each pixel at a given timestep,  $I_i^{target}$  is the original HR image,  $y_i^{(t-1)}$  is the image from the previous timestep, and  $y_i^{(t)}$  is the image from the current timestep [3]. Equation 3.1 reveals how the squared error between each pixel and its target has changed after taking a certain action [3]. If the given agent chooses an action that improves the state, the reward is positive. If the action makes the state worse, the reward is negative. Herein, we attempt to maximize the total reward in Equation 3.1 by minimizing the squared error between each state and the original image. This forces the output image to resemble the original HR image.

It is important to note that the network does not alter the dimensionality of the input image. Therefore, the image being fed through the network has to be pre-processed to match the scale of the desired output. In our work we up-sample by a scaling factor of 4 via bicubic interpolation, which contributes to the novelty of the proposed algorithm as we apply RL to SISR. This initial up-sampling functions as its own action that lays the foundation of the output image. Figure 3.2 illustrates the action of “do nothing” which, for

timestep 1, in effect is choosing to preserve the bicubic pixel value instead of replacing it with the pixel value of another option in the action set.

### 3.3 Experiments and Results

The proposed network is trained on the DIV2K dataset [33], consisting of 800 high definition HR training images. For memory retention and speed, the network is trained on  $60 \times 60$  random crops of the 800 training images. These images are initially blurred using a Gaussian filter, which approximates real-world distortions in the image capture process by simulating a camera’s point spread function [15]. A standard deviation of  $\sigma = 1.3$  is used in each of the images’ spatial dimensions, as per the advice of [19]. After being blurred, they are down-sampled by a factor of 4, interpolated using bicubic interpolation, and immediately up-sampled using bicubic interpolation. Note that the results in Table 3.1 for the four GAN algorithms do not match those in the original papers since the authors did not make any reference to using a Gaussian filter in the creation of their LR images. The up-sampled images, using bicubic interpolation, are fed to the network in the initial timestep. This initial image acts as an action in and of itself because the network can choose to “do nothing”, keeping the bicubic pixel value. The network requires approximately 48 hours to train for 10,000 epochs on a single NVIDIA RTX 2080Ti.

For testing and validating the network, the whole image is sent through the network. Each pixel has its own agent and chooses which action produces the best result. The same preprocessing approach is carried out: blurred, down-sampled, and up-sampled. The testing images come from commonly used testing datasets: Set5, DIV2K, and Urban100. Images 0855, 0878, 0879, and 0891 from DIV2K are excluded from the dataset due to memory issues resulting from their size. All results in Table 3.1 are calculated without these four images

Table 3.1: Quantitative comparison between the proposed method and other competing SR algorithms in RGB color space (Best results in **bold**, second best in *italics*).

Method	Metric	Set5	Urban100	DIV2K Val.
bicubic	MSE	193.46	563.64	209.81
	PSNR	26.69	21.70	26.66
	SSIM	0.7961	0.6442	0.7581
edsr baseline	MSE	<i>105.90</i>	<i>426.18</i>	<i>163.37</i>
	PSNR	<i>28.73</i>	<i>23.14</i>	<i>27.92</i>
	SSIM	<i>0.8631</i>	0.7450	0.8100
esrgan	MSE	250.04	1031.3	507.44
	PSNR	24.61	18.89	22.87
	SSIM	0.7166	0.5569	0.6298
esrgan-psnr	MSE	107.99	457.53	170.42
	PSNR	28.61	22.57	27.67
	SSIM	0.8625	<i>0.7482</i>	<i>0.8139</i>
ppon	MSE	222.01	828.63	437.44
	PSNR	25.21	19.86	23.39
	SSIM	0.7335	0.6213	0.6622
Proposed	MSE	<b>99.563</b>	<b>411.09</b>	<b>159.52</b>
	PSNR	<b>28.97</b>	<b>23.28</b>	<b>28.08</b>
	SSIM	<b>0.8664</b>	<b>0.7517</b>	<b>0.8140</b>
Proposed	MSE	107.83	428.32	165.35
VGG	PSNR	28.65	23.10	27.85
Rewards	SSIM	0.8612	0.7417	0.8044

to maintain a fair comparison. These datasets make it simple to quantitatively compare results with other SISR techniques. We compare results with the following metrics: Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity index (SSIM).

### 3.3.1 VGG Rewards

Perceptual loss in the feature space has previously been explored by a number of sources. Instead of calculating the pixel-by-pixel error between the ground truth image and the generated image, the error is calculated in the feature space. This encourages the the network to generate images that have a similar feature representation to the ground truth images



[34]. Sajjadi et al. extracts the feature representations by sending the HR and SR images through a pre-trained implementation of the VGG-19 network [34]. This network consists of a combination of convolution and max pooling layers that extract features as it decreases the spatial dimension of the image [35]. We extract features from the “conv2.2” layer of Chainer’s VGG-19 implementation. The MSE is calculated between the ground truth and generated feature spaces before being up-sampled via nearest neighbor interpolation to the original crop size of  $60 \times 60$ . This tensor is then added to the total reward calculated with Equation 3.1. The VGG results, in Table 3.1, are only slightly better than the ESRGAN-PSNR images, but present a good quantitative comparison.

### 3.4 Discussion

When GANs were first introduced, they aggressively exceeded existing SISR models. Since the development of so many GANs, their breakthroughs have started to level off. Instead of learning a complex mapping and neglecting the pixel level of an image, the proposed method works on a pixel-by-pixel basis. Therefore, we must zoom in to the pixel level of an image to observe the full effect of the proposed method. Figures 3.4 and 3.5 show a qualitative comparison between sections of two different images. Figure 3.4 shows image `img_005` selected from the Urban100 dataset. This image demonstrates the advantage of using the proposed method for SR. The timesteps manifest the algorithm’s process through visual representations. Looking at these timesteps for the first section, we can see the algorithm chooses mostly EDSR and ESRGAN-PSNR (see Figure 3.2). The algorithm also chooses to increment and decrement the pixel values which helps to increase the PSNR and SSIM values. We can see that EDSR most accurately recreates the section, both qualitatively and quantitatively. All the other options either blur the image or add in

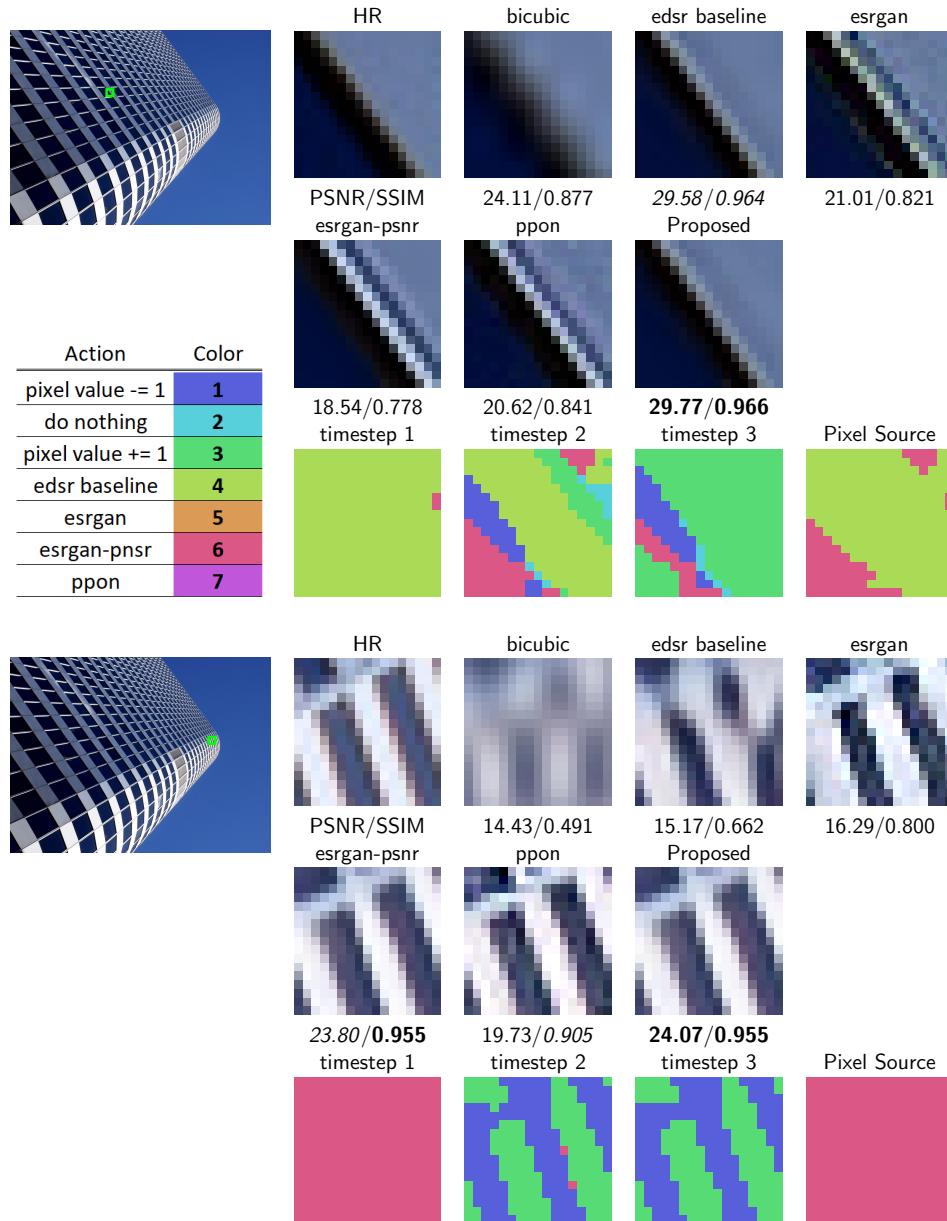


Figure 3.4: Qualitative comparison between the proposed methods and other competing SR algorithms on image img\_005 from the Urban100 dataset (Best results in **bold**, second best in *italics*). The resulting image chooses between multiple options (seen in Pixel Source), where the chosen option is the best. The Pixel Source excludes pixel increment and decrement to show the source image of each pixel. The timestep and Pixel Source images are from the proposed RGB pixel-wise method.

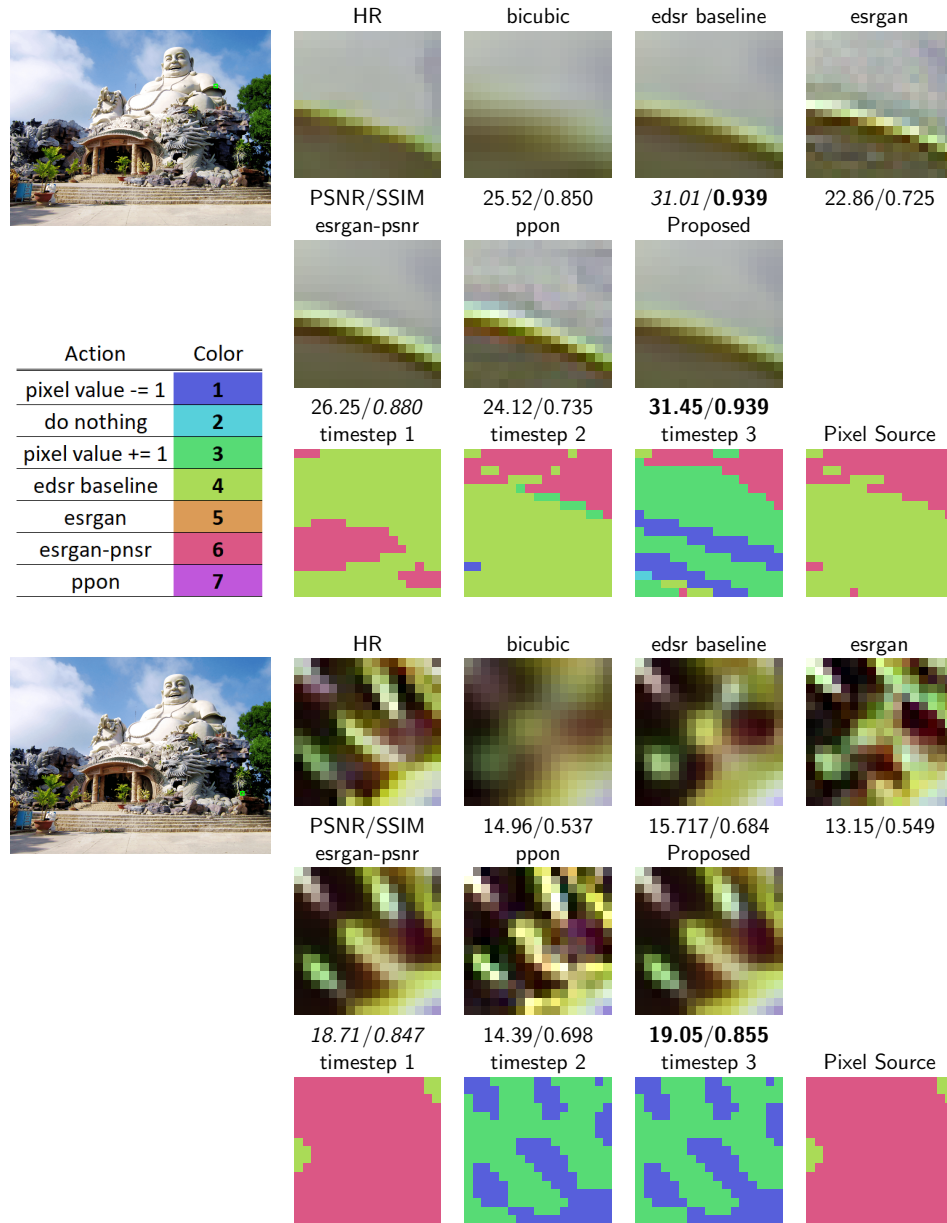


Figure 3.5: Qualitative comparison between the proposed methods and other competing SR algorithms on image 0818 from the DIV2K dataset (Best results in **bold**, second best in *italics*). The resulting image chooses between multiple options (seen in Pixel Source), where the chosen option is the best. The Pixel Source excludes pixel increment and decrement to show the source image of each pixel. The timestep and Pixel Source images are from the proposed RGB pixel-wise method.

artifacts not found in the HR image. Looking at the timesteps for the second section, we can see the algorithm chooses exclusively ESRGAN-PSNR. The algorithm also chooses to increment and decrement the pixel values which helps it to surpass the PSNR and SSIM values of ESRGAN-PSNR. Here we can see that ESRGAN-PSNR most accurately recreates the section, both qualitatively and quantitatively. This illustrates that some SR algorithms are more adequate for different parts of an image. Similarly, Figure 3.5 shows sections of image 0818 from the DIV2K validation dataset. Again, we show that selecting from multiple sources can increase the qualitative and quantitative results of an image. In both figures, the proposed method either obtains the highest metric values or scores the same as one of the options. Consequently, the proposed method learns to quantitatively outperform the best SR algorithms over the entire datasets (see Table 3.1).

We modified the pixelRL problem setting to handle the SISR task. The proposed method learns a strategy to increase the resolution of an image by choosing the best pixel values from each of the GAN options provided. The qualitative and quantitative results support our theory: choosing between multiple GAN options at the pixel level can increase the overall SISR performance. The proposed method’s visual results give us the advantage of observing where each GAN option is most effective. Some options are more effective in low frequency portions of the image, while others are more effective in high frequency portions. This expression can be used to determine which GAN to use in certain scenarios. Other SISR applications can benefit greatly from this visual insight.

## CHAPTER IV

### CONCLUSION

The network from Chapter I uses residual blocks from the ESRGAN [1] and its up-sampling procedure from Hongyang Gao et al. [15]. These two components paired with the residual image addition and pre-training techniques greatly improve the SR performance on IR images. While the residual and upsampling blocks create additional connections between layers, the residual image addition helps with the gradient flow, and the pre-training procedure creates a strong base for the network to start training. The results outperformed the compared upsampling techniques with higher metric values and appearance. The RL network from Chapter II obtained better qualitative and quantitative results than any GAN found in the action set. Therefore, compiling multiple GAN options, at the pixel level, can notably increase the resolution of images. In future work, more residual blocks will be explored, and different loss functions for the proposed RL algorithm will be tested.

## BIBLIOGRAPHY

- [1] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. C. Loy, Y. Qiao, and X. Tang, “ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks,” *arXiv:1809.00219 [cs]*, Sep. 2018, arXiv: 1809.00219. [Online]. Available: <http://arxiv.org/abs/1809.00219>
- [2] H. Gao, H. Yuan, Z. Wang, and S. Ji, “Pixel transposed convolutional networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019.
- [3] R. Furuta, N. Inoue, and T. Yamasaki, “Fully Convolutional Network with Multi-Step Reinforcement Learning for Image Processing,” *arXiv:1811.04323 [cs]*, Nov. 2018, arXiv: 1811.04323. [Online]. Available: <http://arxiv.org/abs/1811.04323>
- [4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680. [Online]. Available: <http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf>
- [5] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, M. Hasan, B. C. V. Esesn, A. A. S. Awwal, and V. K. Asari, “The history began from alexnet: A comprehensive survey on deep learning approaches,” *CoRR*, vol. abs/1803.01164, 2018. [Online]. Available: <http://arxiv.org/abs/1803.01164>
- [6] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4681–4690.
- [7] X. Zhang, C. Li, Q. Meng, S. Liu, Y. Zhang, and J. Wang, “Infrared image super resolution by combining compressive sensing and deep learning,” *Sensors*, vol. 18, no. 8, p. 2587, 2018.
- [8] Y. Liu, X. Chen, J. Cheng, H. Peng, and Z. Wang, “Infrared and visible image fusion with convolutional neural networks,” *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 16, no. 03, p. 1850018, 2018.
- [9] J. Zhong, B. Yang, Y. Li, F. Zhong, and Z. Chen, “Image fusion and super-resolution with convolutional neural network,” in *Chinese Conference on Pattern Recognition*. Springer, 2016, pp. 78–88.
- [10] X. Kuang, X. Sui, Y. Liu, Q. Chen, and G. Guohua, “Single infrared image optical noise removal using a deep convolutional neural network,” *IEEE Photonics Journal*, vol. 10, no. 2, pp. 1–15, 2018.

- [11] J. Ma, W. Yu, P. Liang, C. Li, and J. Jiang, “FusionGAN: A generative adversarial network for infrared and visible image fusion,” *Information Fusion*, vol. 48, pp. 11–26, 08 2019.
- [12] *Deep generative adversarial networks for infrared image enhancement*, vol. 10661, 2018. [Online]. Available: <https://doi.org/10.1117/12.2304875>
- [13] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” *arXiv preprint arXiv:1802.05957*, 2018.
- [14] A. Jolicœur-Martineau, “The relativistic discriminator: a key element missing from standard GAN,” *CoRR*, vol. abs/1807.00734, 2018. [Online]. Available: <http://arxiv.org/abs/1807.00734>
- [15] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, “Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [16] M. Bosch, C. M. Gifford, and P. A. Rodriguez, “Super-resolution for overhead imagery using densenets and adversarial learning,” in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 1414–1422.
- [17] Y. Zhang and Y. Xiang, “Recent advances in deep learning for single image super-resolution,” in *Advances in Brain Inspired Cognitive Systems*, J. Ren, A. Hussain, J. Zheng, C.-L. Liu, B. Luo, H. Zhao, and X. Zhao, Eds. Cham: Springer International Publishing, 2018, pp. 85–95.
- [18] Y. Sun, L. Li, P. Cong, Z. Wang, and X. Guo, “Enhancement of digital radiography image quality using a convolutional neural network,” *Journal of X-ray Science and Technology*, vol. 25, no. 6, pp. 857–868, 2017.
- [19] C.-Y. Yang, C. Ma, and M.-H. Yang, “Single-image super-resolution: A benchmark,” in *European Conference on Computer Vision*. Springer, 2014, pp. 372–386.
- [20] G. Perarnau, J. van de Weijer, B. Raducanu, and J. M. Álvarez, “Invertible conditional gans for image editing,” *CoRR*, vol. abs/1611.06355, 2016. [Online]. Available: <http://arxiv.org/abs/1611.06355>
- [21] H. Zhang, V. Sindagi, and V. M. Patel, “Image de-raining using a conditional generative adversarial network,” *arXiv preprint arXiv:1701.05957*, 2017.
- [22] H. Shi, J. Dong, W. Wang, Y. Qian, and X. Zhang, “SSGAN: secure steganography based on generative adversarial networks,” *CoRR*, vol. abs/1707.01613, 2017. [Online]. Available: <http://arxiv.org/abs/1707.01613>

- [23] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, “Learning from simulated and unsupervised images through adversarial training,” *CoRR*, vol. abs/1612.07828, 2016. [Online]. Available: <http://arxiv.org/abs/1612.07828>
- [24] J. Pan, C. Canton-Ferrer, K. McGuinness, N. E. O’Connor, J. Torres, E. Sayrol, and X. Gir’o i Nieto, “Salgan: Visual saliency prediction with generative adversarial networks,” *CoRR*, vol. abs/1701.01081, 2017. [Online]. Available: <http://arxiv.org/abs/1701.01081>
- [25] Z. Li and X. Zhang, “Collaborative Deep Reinforcement Learning for Image Cropping,” in *2019 IEEE International Conference on Multimedia and Expo (ICME)*, Jul. 2019, pp. 254–259.
- [26] Q. Cao, L. Lin, Y. Shi, X. Liang, and G. Li, “Attention-Aware Face Hallucination via Deep Reinforcement Learning,” *arXiv:1708.03132 [cs]*, Aug. 2017, arXiv: 1708.03132. [Online]. Available: <http://arxiv.org/abs/1708.03132>
- [27] W. Li, X. Feng, H. An, X. Y. Ng, and Y.-J. Zhang, “Mri reconstruction with interpretable pixel-wise operations using reinforcement learning,” in *AAAI*, 2020.
- [28] X. Liao, W. Li, Q. Xu, X. Wang, B. Jin, X. Zhang, Y. Zhang, and Y. Wang, “Iteratively-Refined Interactive 3d Medical Image Segmentation with Multi-Agent Reinforcement Learning,” *arXiv:1911.10334 [cs]*, Nov. 2019, arXiv: 1911.10334. [Online]. Available: <http://arxiv.org/abs/1911.10334>
- [29] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous Methods for Deep Reinforcement Learning,” *arXiv:1602.01783 [cs]*, Feb. 2016, arXiv: 1602.01783. [Online]. Available: <http://arxiv.org/abs/1602.01783>
- [30] K. Cho, B. van Merriënboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” *CoRR*, vol. abs/1406.1078, 2014. [Online]. Available: <http://arxiv.org/abs/1406.1078>
- [31] Z. Hui, J. Li, X. Gao, and X. Wang, “Progressive Perception-Oriented Network for Single Image Super-Resolution,” *arXiv:1907.10399 [cs, eess]*, Jul. 2019, arXiv: 1907.10399. [Online]. Available: <http://arxiv.org/abs/1907.10399>
- [32] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, “Enhanced Deep Residual Networks for Single Image Super-Resolution,” *arXiv:1707.02921 [cs]*, Jul. 2017, arXiv: 1707.02921. [Online]. Available: <http://arxiv.org/abs/1707.02921>
- [33] E. Agustsson and R. Timofte, “Ntire 2017 challenge on single image super-resolution: Dataset and study,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.



- [34] M. S. M. Sajjadi, B. Schölkopf, and M. Hirsch, “Enhancenet: Single image super-resolution through automated texture synthesis,” *CoRR*, vol. abs/1612.07919, 2016. [Online]. Available: <http://arxiv.org/abs/1612.07919>
- [35] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.