



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

**Deep Learning approaches for Robotic  
Grasp Detection and Image  
Super-Resolution**

**Dongwon Park**

**Department of Electrical and Engineering  
Graduate School of UNIST**


# Deep Learning approaches for Robotic Grasp Detection and Image Super-Resolution

A thesis  
submitted to the Graduate School of UNIST  
in partial fulfillment of the  
requirements for the degree of  
Master of Science

Dongwon Park

May 30, 2019

Approved by




Major Advisor  
Se Young Chun

# Deep Learning approaches for Robotic Grasp Detection and Image Super-Resolution

Dongwon Park

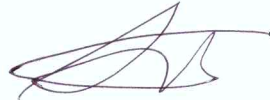
This certifies that the thesis of Dongwon Park is approved.

May 30, 2019



---

Advisor: Se Young Chun



---

Sung-Phil Kim: Thesis Committee Member #1



---

Jae-Sik Choi: Thesis Committee Member #2

## Abstract

In recent years, many papers mentioned that use Deep learning to objects detection and robot grasping detection have improved accuracy with higher image resolutions. We use the Deep learning to describe robot grasp detection and image super-resolution related two papers.

### **0.0.1 Real-Time, Highly Accurate Robotic Grasp Detection using Fully Convolutional Neural Networks with High-Resolution Images**

Robotic grasp detection for novel objects is a challenging task, but for the last few years, deep learning based approaches have achieved remarkable performance improvements, up to 96.1% accuracy, with RGB-D data. In this paper, we propose fully convolutional neural network (FCNN) based methods for robotic grasp detection. Our methods also achieved state-of-the-art detection accuracy (up to 96.6%) with state-of-the-art real-time computation time for high-resolution images (6-20ms per  $360 \times 360$  image) on Cornell dataset. Due to FCNN, our proposed method can be applied to images with any size for detecting multigrasps on multiobjects. Proposed methods were evaluated using 4-axis robot arm with small parallel gripper and RGB-D camera for grasping challenging small, novel objects. With accurate vision-robot coordinate calibration through our proposed learning-based, fully automatic approach, our proposed method yielded 90% success rate.

### **0.0.2 Efficient Module Based Single Image Super Resolution for Multiple Problems**

Example based single image super resolution (SR) is a fundamental task in computer vision. It is challenging, but recently, there have been significant performance improvements using deep learning approaches. In this article, we propose efficient module based single image SR networks (EMBSR) and tackle multiple SR problems

in NTIRE 2018 challenge by recycling trained networks. Our proposed EMBSR allowed us to reduce training time with effectively deeper networks, to use modular ensemble for improved performance, and to separate subproblems for better performance. We also proposed EDSR-PP, an improved version of previous ESDR by incorporating pyramid pooling so that global as well as local context information can be utilized. Lastly, we proposed a novel denoising / deblurring residual convolutional network (DnResNet) using residual block and batch normalization. Our proposed EMBSR with DnResNet demonstrated that multiple SR problems can be tackled efficiently and effectively by winning the 2nd place for Track 2 and the 3rd place for Track 3. Our proposed method with EDSR-PP also achieved the ninth place for Track 1 with the fastest run time among top nine teams.

## Contents

0.0.1	Real-Time, Highly Accurate Robotic Grasp Detection using Fully Convolutional Neural Networks with High-Resolution Images . . . . .	iv
0.0.2	Efficient Module Based Single Image Super Resolution for Multiple Problems	iv
<b>Contents</b>		<b>vi</b>
<b>List of Figures</b>		<b>viii</b>
<b>List of Tables</b>		<b>x</b>
<b>I.</b>	<b>Introduction</b>	<b>1</b>
1.0.1	Real-Time, Highly Accurate Robotic Grasp Detection using Fully Convolutional Neural Networks with High-Resolution Images . . . . .	2
1.0.2	Efficient Module Based Single Image Super Resolution for Multiple Problems	4
<b>II.</b>	<b>Real-Time, Highly Accurate Robotic Grasp Detection using Fully Convolutional Neural Networks with High-Resolution Images</b>	<b>8</b>
2.1	Background and Related Works . . . . .	8
2.2	PROPOSED METHODS FOR ROBOTIC GRASPS . . . . .	10
2.2.1	Problem Description . . . . .	10
2.2.2	Reparametrization of 5D Grasp Representation and Grasp Probability . .	11
2.2.3	Loss Function for Robotic Grasp Detection . . . . .	12
2.2.4	Proposed FCNN Architecture . . . . .	13
2.2.5	Learning-based Vision-Robot Calibration . . . . .	14
2.3	EXPERIMENTS AND EVALUATION . . . . .	16
2.3.1	Evaluation with Cornell Dataset . . . . .	16
2.3.2	Evaluation with 4-axis Robot Arm and RGB-D . . . . .	17
2.4	RESULTS . . . . .	17
2.4.1	Evaluation Results on Cornell Dataset . . . . .	17
2.4.2	Evaluation Results with 4-Axis Robot Arm . . . . .	18

---

<b>III.</b>	<b>Efficient Module Based Single Image Super Resolution for Multiple Problems</b>	<b>21</b>
3.1	Background and Related Works . . . . .	21
3.2	Method . . . . .	22
3.2.1	Modular Approach . . . . .	22
3.2.2	SR Module: EDSR-PP (Pyramid Pooling) . . . . .	25
3.2.3	Denoising / Deblurring Module: DnResNet . . . . .	26
3.3	Experiment . . . . .	28
3.3.1	Dataset . . . . .	28
3.3.2	Training and Alignment . . . . .	28
3.3.3	DIV2K Validation Set Results . . . . .	28
3.3.4	Results of NTIRE 2018 Challenge . . . . .	29
<b>IV.</b>	<b>Conclusion</b>	<b>34</b>
	<b>References</b>	<b>35</b>



## List of Figures

1.1	Camera–robot configurations used in robot grasping detection: (a) monocular eye–in–hand, (b) monocular stand–alone, (c) monocular stand–alone. . . . .	2
1.2	(Left) an example of detecting multiple robotic grasps (5D grasp representations) for multiple objects in one image using our proposed method. (Right) an example of our real robotic grasp experiment picking up a toothbrush. . . . .	3
1.3	An example of given images for NTIRE 2018 challenge on image super-resolution. The goal of challenge was to design algorithms to map from low resolution images (Classic bicubic $\times 8$ , Mild adverse condition $\times 4$ or Difficult adverse condition $\times 4$ ) to a high resolution image (HR). . . . .	5
1.4	(a) Module based approach for Track 1 SR problem. (b) Module based approach for Tracks 2, 3 SR problems. The solution for module problem (B) can be efficiently recycled among different SR problems in all Tracks. . . . .	6
2.1	A typical multibox approach for robotic grasp detection. An input image is divided into $S \times S$ grid and regression based robotic grasp detection is performed on each grid box. Then, the output with the highest grasp probability is selected as the final result. This approach can be applied to multiobject, multigrasp detection tasks. . . . .	9
2.2	(a) A 5D grasp representation with location $(x, y)$ , orientation $\theta$ , gripper opening width $w$ and plate size $h$ . (b) For the $(2, 2)$ grid cell, all parameters for 5D grasp representation are illustrated including a pre-defined anchor box (black dotted box), a 5D grasp representation (blue box). . . . .	11
2.3	Proposed FCNN architecture based on Darknet-19. . . . .	14
2.4	Proposed learning-based vision-robot calibration. . . . .	15
2.5	Calibration error (in mm) for $x, y$ in robot coordinate system over increasing number of learning samples. . . . .	16
2.6	Images from Cornell grasp detection dataset. . . . .	17
2.7	Novel objects for real robot grasping tasks. . . . .	18

**LIST OF FIGURES**

---

2.8	One grasp detection results with different image resolution, data type, and with different deep network. All methods were able to detect large grasp areas, but the methods with small deep network and/or low image resolution missed some small grasp areas. . . . .	19
2.9	An illustration of our robot grasp experiment with “candy” (Left) and multigrasp detection results for “candy” using 4 different methods. Ours (360) successfully detect stick part of the candy. . . . .	20
3.1	Modular approach for multiple SR problems. Among 9 modules, 5 modules required long training while 4 modules can be recycled with short fine tuning. . . .	23
3.2	An illustration of our proposed EDSR-PP. Upsampling lay of the original EDSR [1] was replaced with pyramid pooling structure. . . . .	25
3.3	An illustration of our proposed DnResNet. Unlike DnCNN that uses CNN layers [2], residual blocks (Resblock) were used as a basic building block. . . . .	26
3.4	Comparison of residual blocks for SRResNet [3], EDSR [1], and our DnResNet. . . .	27
3.5	SR results of Track 1 in NTIRE 2018 challenge (bicubic downsampling $\times 8$ ). Our EMBSR yielded better PSNR and slightly sharper images than EDSR. . . . .	30
3.6	SR results of Track 2 in NTIRE 2018 challenge (unknown downsampling $\times 4$ with mild blur and noise). Our EMBSR yielded superior PSNR and image quality to EDSR and efficiently tackled SR problem with mild image degradation. . . . .	31
3.7	SR results of Track 3 in NTIRE 2018 challenge (unknown downsampling $\times 4$ with mild blur and noise). Our EMBSR yielded superior PSNR and image quality to EDSR. EDSR does not seem to deal with multiple problems (SR, denoising, deblurring) well while our EMBSR efficiently tackled SR problem with multiple sources of image degradation. . . . .	32

## List of Tables

2.1	Performance summary on the Cornell dataset with IOU metric. Our proposed methods yielded state-of-the-art prediction accuracy in both image-wise and object-wise splits with state-of-the-art computation time. Note that Resnet-50, Darknet-19, Alexnet require 82.6, 48.5, and 6.0MB memory, respectively. Performance unit is in % unless specified. . . . .	18
2.2	Performance summary of real robotic grasping for 6 novel, small objects with 5 repetitions. For Lenz and Redmon, our in-house implementations (modifications) were used after validating their performance with the Cornell dataset. Darknet implementation was used for Ours with resized image (224) and with high resolution image (360). . . . .	19
3.1	PSNR (dB) results of different methods for DIV2K validation data set: SRCNN [4], VDSR [5], EDSR [1], and our proposed EMBSR. . . . .	29
3.2	Performance comparison between architectures on the DIV2K validation set (PSNR in dB). . . . .	29
3.3	Preliminary results of NTIRE 2018 challenge, Track 1, $\times 8$ bicubic downsampling (PSNR in dB). . . . .	29
3.4	Preliminary results of NTIRE 2018 challenge, Track 2, $\times 4$ unknown downsampling with mild blur and noise (PSNR in dB). . . . .	30
3.5	Preliminary results of NTIRE 2018 challenge, Track 3, $\times 4$ unknown downsampling with difficult blur and noise (PSNR in dB). . . . .	33

---

## Acknowledgement

---

I cannot believe that it has been already 2 years 6 months since I studied for a master's degree. Without so many people around me, this thesis may not have been completed. First of all, I would like to thank my supervisor Professor Se Young Chun for his guidance and teaching throughout my study and research. If he did not offer me as a master student with this research, I may not achieve these great works and experiences. I also want to express my sincere thanks to my defense committee members: Professor Sung-Phil Kim, for providing the story line comments and encouraging me to understand the level, and Professor Jae-Young Sim for his comments about experimental results description. I would also like to acknowledge my lab mates: Hanvit Kim, Thanh Quoc Phan, Magaiya Zhussip, Shakarim Soltanayev, Ji-Soo Kim, Kwan-Young Kim, Won Jae Hong, DongUn Kang, Yong Hyeok Seo, Haesoo Eun and Byung-Hyun Lee. Lastly, I would like to give my very special thanks to my family for always believing in me, especially my parents, who always encouraged me whenever I was down.

---

# Introduction

---

In recent years, Deep learning has produced good results in computer vision. such as image super-resolution, object detection, robot grasping and image denoising and so on. In robot grasping, vision is essential because it recognizes objects and looks for grasping points. Robot gripper and camera should be used at the same time. So, we have to think a lot about the camera position. As shown in Figure 1 (a), when we install the camera on the robot gripper, local areas can be seen in detail, but global areas are hard to see. Also, to install the camera on the gripper, we should use a small size camera. For this reason, we use a low quality camera with a low resolution. As shown in Figure 1 (b) and (c), when the camera is installed outside, it is possible to see a global area, but there is a disadvantage that it can not be seen in detail compared to (a). This problem can be solved by using Image super resolution. In recent years, many papers mentioned that use Deep learning to objects detection and robot grasping detection have improved accuracy with higher image resolutions. The goal of image super resolution (SR) problem is to design an algorithm to map from low resolution(LR) images to a high resolution (HR) image. Because of this, through Image SR, we can efficiently use low resolution camera in robot grasping detection, and even if we install the camera outside, we can detect more detail. So we did pre-research on robot grasp detection topic and image super-resolution topic.

In this paper, We have described two articles. The second paper describes the results of FCNN based robot grasping according to image resolution and how to calibrate robot and camera. The third article describes EMBSR that can be efficiently solved in multiple problems.

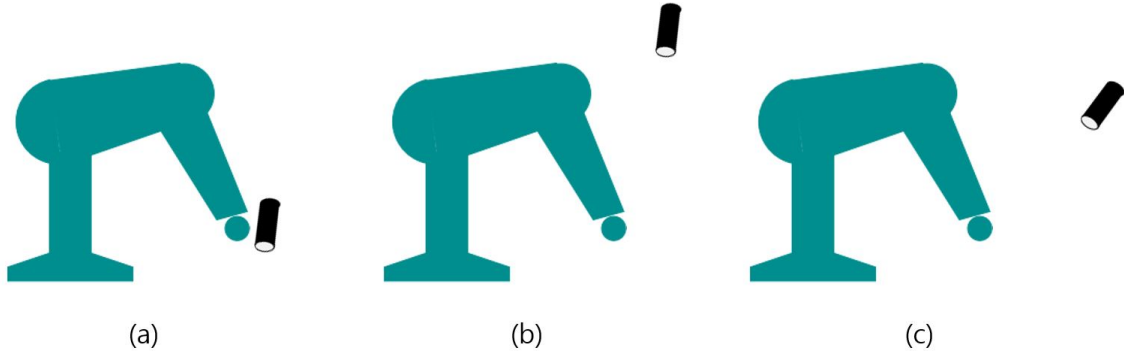


Figure 1.1: Camera-robot configurations used in robot grasping detection: (a) monocular eye-in-hand, (b) monocular stand-alone, (c) monocular stand-alone.

### 1.0.1 Real-Time, Highly Accurate Robotic Grasp Detection using Fully Convolutional Neural Networks with High-Resolution Images

Robot grasping of novel objects has been investigated extensively, but it is still a challenging, open problem in robotics. Humans instantly identify multiple grasping areas of novel objects (perception) and almost instantly plan how to pick them up (planning), and then actually grasp it reliably (control). However, accurate robotic grasp detection, trajectory planning, and reliable execution are quite challenging for robots. As the first step, detecting robotic grasps accurately and quickly from imaging sensors (*e.g.*, RGB-D camera) is an important task for successful robotic grasping.

Robotic grasp detection or synthesis has been widely investigated for many years. Grasp synthesis is divided into analytical and empirical (or data-driven) methods [6] for known, familiar objects and novel objects [7]. In particular, machine learning (non-deep learning) based approaches for robotic grasp detection have utilized data to learn discriminative features for a suitable grasp configuration and to yield excellent performance on generating grasp locations [8–10]. A typical approach for them is to use a sliding window to select local image patches and to evaluate graspability so that the best image patch with the highest graspability score is chosen for robotic grasp detection result. In 2011, one of the state-of-the-art graspability prediction accuracies without deep learning was 60.5% and its computation time per image was very slow due to sliding windows (50 sec per image) [10].

Deep learning has been successful in computer vision applications such as image classification [11, 12] and object detection [13, 14]. Deep learning has also been utilized for robotic grasp detection and has achieved significant improvements over conventional methods. Lenz *et al.* proposed deep learning classifier based robotic grasp detection methods that achieved up to 73.9% (image-wise) and 75.6% (object-wise) prediction accuracy [15, 16]. However, its computation time per image was still slow (13.5 sec per image) due to sliding windows. Redmon *et*

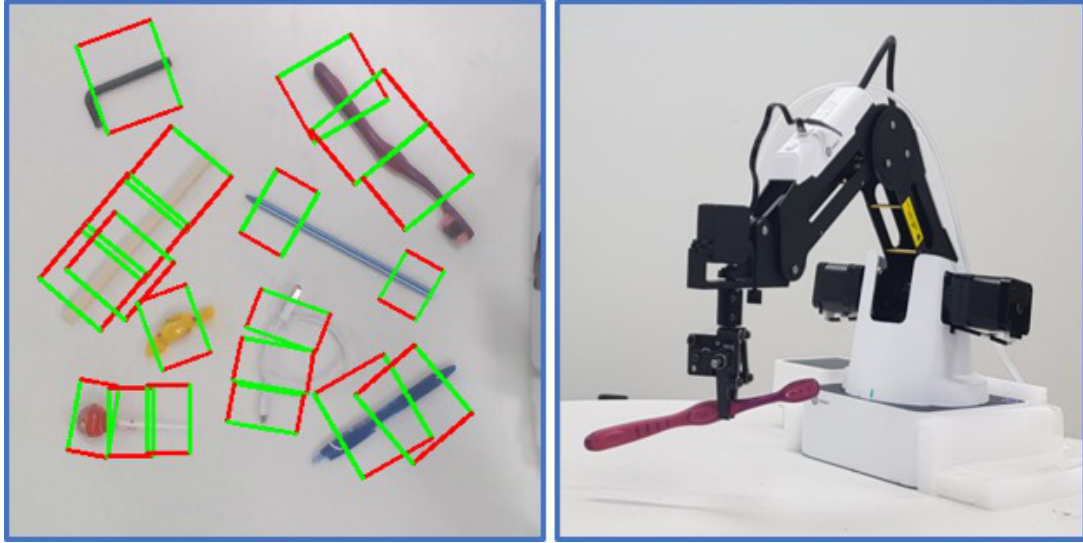


Figure 1.2: (Left) an example of detecting multiple robotic grasps (5D grasp representations) for multiple objects in one image using our proposed method. (Right) an example of our real robotic grasp experiment picking up a toothbrush.

*al.* proposed deep learning regressor based grasp detection methods that yielded up to 88.0% (image-wise) and 87.1% (object-wise) with remarkably fast computation time (76 ms per image) [17]. Recently, Chu *et al.* proposed two-stage neural networks with grasp region proposal network and robotic grasp detection networks and have achieved up to 96.0% (image-wise) and 96.1% (object-wise) prediction accuracies [18]. However, its computation time has slightly increased due to region proposal network (120 ms per image). Real-time robotic grasp detection can be critical for some applications with dynamic environment or dynamic objects. Thus, reducing computation time while maintaining high prediction accuracy seems desirable.

In this paper, we proposed novel fully convolutional neural network (FCNN) based methods for robotic grasp detection. Our proposed methods yielded state-of-the-art performance comparable to the work of Chu *et al.* [18] while their computation time is much faster for high resolution image ( $360 \times 360$  image). Note that most deep learning based robotic grasp detection works used  $227 \times 227$  resized image including [18]. Our proposed methods can perform multiobject, multigrasp detection as shown in Fig. 1.2 (Left). Our proposed methods were evaluated with a 4-axis robot as shown in Fig. 1.2 (Right) and achieved 90% success rate for real grasping tasks with novel objects. Since this small robot has a gripper with the maximum range of 27.5 mm, it was critical to accurately calibrate robotic grasp information and our vision system information. We proposed a simple learning-based vision-robot calibration method and achieved accurate calibration and robot grasping performance. Here is the summary of the contributions of this paper:

1. Newly proposed real-time, single-stage FCNN based robotic grasp detection methods that yielded state-of-the-art computation time for high resolution image ( $360 \times 360$  image) while achieving comparable state-of-the-art prediction accuracies, especially for more strict performance metrics. For example, our method achieved 96.6% image-wise, 95.1% object-wise with 10 ms per high-resolution image while the work of Chu *et al.* [18] achieved 96.0% image-wise, 96.1% object-wise with 120 ms per low-resolution image. In other words, our method yielded comparable accuracies with  $12 \times$  faster computation than Chu *et al.* [18]. Our FCNN based methods can be applied to multigrasp, multiobject detection.
2. Our proposed methods were evaluated for real grasping tasks and yielded 90.0% success rate with challenging small, novel objects and with a small parallel gripper (max open width 27.5 mm). This was possible due to our proposed simple, full automatic learning-based approach for vision-robot calibration. Our method achieved less than 1.5 mm error for calibration, which is close to vision resolution.

### 1.0.2 Efficient Module Based Single Image Super Resolution for Multiple Problems

The goal of image super resolution (SR) problem is to design an algorithm to map from low resolution (LR) image(s) to a high resolution (HR) image. Conventional SR was to yield a HR image from a multiple of LR images (*e.g.*, video) considering a number of LR image degradation operators such as blurring and noise. This type of SR has been well studied [19] and fundamental performance limit for it has been analyzed [20]. In medical imaging, generating a high signal-to-noise ratio (SNR) image from a multiple of low SNR images has also been well studied with similar model based approaches as conventional SR problems [21].

In contrast, a SR problem using a single LR image is challenging since high frequency information in a HR image is lost or degraded due to aliasing during sampling process. Because there was no effective way to extrapolate high frequency information, single image SR problem was usually considered as an interpolation problem [19]. An example based SR method was proposed based on Bayesian belief propagation [22] and a patch based SR method was proposed by combining a conventional multiple image based SR and an example based SR [23].

Deep neural network has applied to many image processing and computer vision problems and has shown significantly improved performance over conventional methods [24]. There have been several works on single image SR problems and several deep neural networks were proposed such as SRCNN [4], VDSR [5], SRResNet [3], and EDSR [1]. EDSR achieved state-of-the-art performance for  $\times 4$  SR problem in terms of peak SNR (PSNR) and structural similarity index



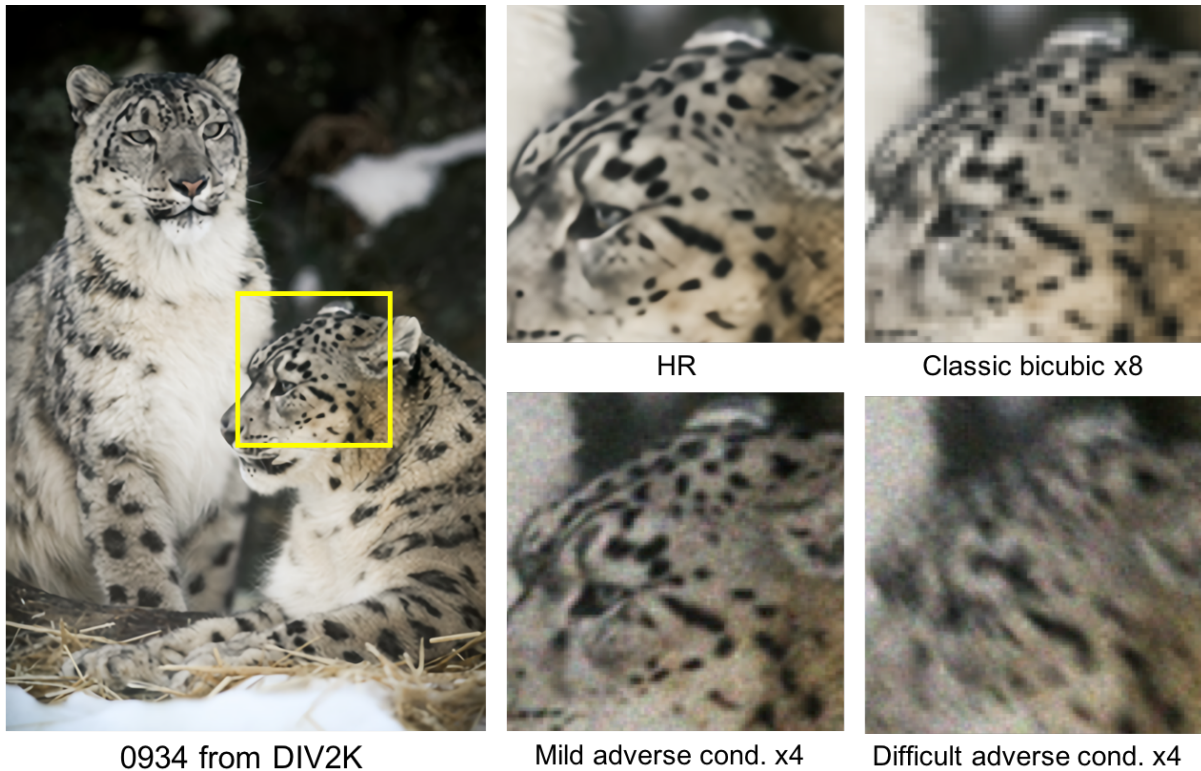


Figure 1.3: An example of given images for NTIRE 2018 challenge on image super-resolution. The goal of challenge was to design algorithms to map from low resolution images (Classic bicubic  $\times 8$ , Mild adverse condition  $\times 4$  or Difficult adverse condition  $\times 4$ ) to a high resolution image (HR).

(SSIM) and won the NTIRE 2017 challenge [25] for SR problems. NTIRE 2017 consisted of two Tracks for known (bicubic) and unknown blurs and for each Track, there were three different downsampling rates ( $\times 2$ ,  $\times 3$ ,  $\times 4$ ). EDSR outperformed other previous networks including SRResNet for all public dataset including DIV2K, NTIRE 2017’s new dataset [1].

NTIRE 2018 is more challenging than its previous challenge by having 4 Tracks: Track 1 with  $\times 8$  SR problem and with known blur (bicubic) and Tracks 2, 3, 4 with  $\times 4$  SR problems and with mild to severe noise and/or unknown blur. Figure 1.3 shows examples of given images for the ground truth and for Tracks 1, 2, 3 that our team participated in. Mild noise was observed in given  $\times 4$  downsampled images for Track 2 and similar level of noise was observed in given  $\times 4$  downsampled images, but with relatively severe unknown blur for Track 3.

In this article, we propose an efficient module based approach for tackling multiple SR problems in Tracks 1, 2, 3 of NTIRE 2018. We decomposed the original problems in Tracks 1, 2, 3 into subproblems as shown in Figures 1.4 (a) (Track 1) and 1.4 (b) (Tracks 2, 3), identified state-of-the-art methods for subproblems as baselines, and efficiently recycled trained deep

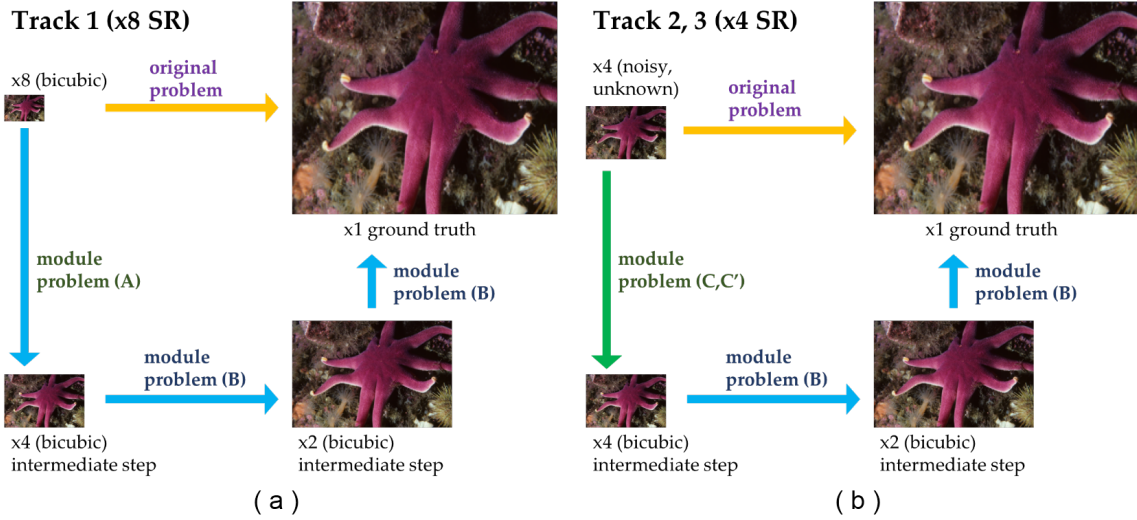


Figure 1.4: (a) Module based approach for Track 1 SR problem. (b) Module based approach for Tracks 2, 3 SR problems. The solution for module problem (B) can be efficiently recycled among different SR problems in all Tracks.

neural networks for subproblems among all problems in different Tracks. Utilizing intermediate goals for  $\times 8$  SR is not new [26] and solving multiple problems together for efficiency is not a new concept [27]. This approach could also be sub-optimal in terms of the overall cost function optimization. However, our proposed method is different from previous works in 1) module based training scheme to save training time for entire networks for Tracks 1, 2, 3 by recycling and to use effectively deeper convolutional networks with more feature map channels in the midst of limited computation and memory resource, in 2) ensemble output of each module for each subproblem to improve the performance further without increasing the complexity of networks, and in 3) separating the problem of SR (increasing the resolution) from the problem of denoising and deblurring (Tracks 2, 3).

We also proposed new deep neural networks to improve the performance for subproblems. For SR problems in module problems (A) and (B) shown in Figures 1.4 and 1.4, EDSR [1] was chosen as our baseline network. In this article, we proposed EDSR-PP by adding pyramid pooling layers [28] to EDSR for further performance improvement with DIV2K dataset. For denoising and deblurring problems in module problem (C, C') as illustrated in Figure 1.4, we adopt DnCNN [2], one of the state-of-the-art methods for denoising and deblurring problem, as our baseline network. We proposed a novel denoising and deblurring network called DnResNet based on residual block structure [29] and showed significant performance improvement over the baseline DnCNN.

Our models were trained using DIV2K training dataset [25] and were evaluated with DVI2K

validation and test dataset. In NTIRE 2018 challenge, our proposed methods won the 2nd place (out of 18 teams) for Track 2 and the 3rd place (out of 18 teams) for Track 3 with our proposed DnResNet and demonstrated that our proposed module based approach can efficiently and effectively solve multiple problems. Our proposed method with our EDSR-PP also achieved the ninth place (out of 24 teams) for Track 1 with the fastest run time among top nine teams. Here is the summary of this article's contributions:

- Modular approach for efficient training with effectively deeper network, improved performance with modular ensemble, and novel problem decomposition.
- EDSR-PP: improved EDSR with pyramid pooling.
- DnResNet: novel architecture for denoising / deblurring based on residual block.

---

# Real-Time, Highly Accurate Robotic Grasp Detection using Fully Convolutional Neural Networks with High-Resolution Images

---

## 2.1 Background and Related Works

**Pre-deep learning era.** Data-driven robotic grasp detection for novel object has been investigated extensively [7]. Saxena *et al.* proposed a machine learning based method to rank the best graspable location for all candidate image patches from different locations [8]. Jiang *et al.* proposed a 5D robotic grasp representation and further improved the work of Saxena *et al.* by proposing a machine learning method to rank the best graspable image patch whose representation includes orientation and gripper distance among all candidates [10]. The work of Jiang *et al.* achieved the prediction accuracy of 60.5% (image-wise) and 58.3% (object-wise) with computing time of 50 sec (50,000 ms) per image.

**Two-stage, classification based approach.** Lenz *et al.* proposed to use a sparse auto-encoder (SAE), an early deep learning model, to rank the best graspable candidate image patch from sliding window with multi-modal information (color, depth and surface norm) [15, 16]. Their methods achieved up to 73.9% (image-wise) and 75.6% (object-wise) prediction accuracy, but its computation time per image was still slow (13.5 sec or 13,500 ms per image) due to time-consuming sliding windows. Wang *et al.* proposed a real-time classification based grasp

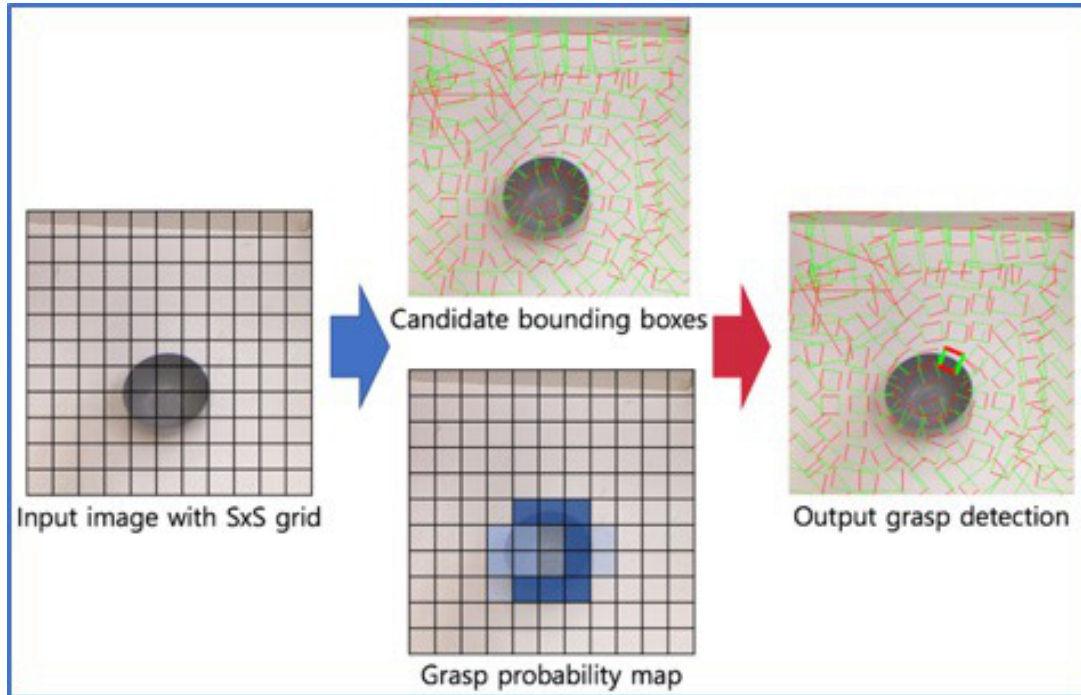


Figure 2.1: A typical multibox approach for robotic grasp detection. An input image is divided into  $S \times S$  grid and regression based robotic grasp detection is performed on each grid box. Then, the output with the highest grasp probability is selected as the final result. This approach can be applied to multiobject, multigrasp detection tasks.

detection method using a stacked SAE for classification, which is similar to the work of Lenz *et al.*, but with remarkably efficient grasp candidates generation [30]. This method utilized prior information and pre-processing to reduce the search space of grasp candidates such as object recognition result and the graspability of previously evaluated image patches. It also reduced the number of grasp representation parameters such as height ( $h$ ) for known gripper and orientation ( $\theta$ ) that could be analytically obtained from surface norm. Mahler *et al.* proposed Dex-Net 2.0 for point clouds based on two-stage approach with GQ-CNN and reported that 93.0% (image-wise) prediction accuracy was achieved [31]. Note that this approach is similar to those of R-CNN [32] or fast R-CNN [33] in object detection.

**Single-stage, regression based approach.** Redmon *et al.* proposed a deep learning regressor based robotic grasp detection method based on the AlexNet [11] that yielded 84.4% (image-wise) and 84.9% (object-wise) with fast computation time (76 ms per image) [17]. When performing robotic grasp regression and object classification together, image-wise prediction accuracy of 85.5% was able to be achieved without increasing computation time. Kumra *et al.* also proposed a real-time regression based grasp detection method using ResNet [12] especially for multimodal information (RGB-D). Their method yielded up to 89.2% (image-wise) and 88.9% (object-wise) prediction accuracies with fast computation time (103 ms per image) [34].

**Multibox based approach.** Redmon *et al.* also proposed a multibox based robotic grasp detection method (called MultiGrasp) by dividing the whole input image into  $S \times S$  grid and applying regression based robotic grasp detection to each grid box [17]. This approach did not increase computation time (76 ms per image), but did increase prediction accuracy up to 88.0% (image-wise) and 87.1% (object-wise). The pipeline of multibox based approach is illustrated in Fig. 2.1. Note that the last step (red arrow) is a simple selection on the highest grasp probability. Simply modifying this last step to select more than one result could result in multiobject, multigrasp detection. Guo *et al.* proposed a hybrid multibox based approach with visual and tactile data based on ZF-net [35] by classifying graspability, orientations ( $\theta$ ), and by regressing locations and graspable width ( $w$ ), height ( $h$ ) [36]. The work of Guo *et al.* achieved 93.2% (image-wise) and 89.1% (object-wise) prediction accuracies.

Note that MultiGrasp by Redmon *et al.* has influenced several object detection methods such as YOLO [37], SSD [38], and recently YOLO9000 [14]. YOLO is based on AlexNet [11] to estimate the location and class of multiple objects [37]. SSD further developed regression based object detection by incorporating intermediate CNN features [38]. Recently, YOLO9000 extended the original YOLO significantly with fast computation and high accuracy [14]. Our proposed robotic grasp detection methods are inspired by YOLO9000 [14].

**Hybrid approach.** Recently, Asif *et al.* proposed GraspNet that predicts graspability and then estimates robotic grasp parameters based on high-resolution grasp probability map [39]. This approach achieved 90.6% (image-wise) and 90.2% (object-wise) with state-of-the-art computation time (24 ms per image). Chu *et al.* proposed two-stage neural networks combining grasp region proposal network and robotic grasp detection network [18] based on Faster R-CNN for object detection tasks [13]. This approach has yielded state-of-the-art prediction accuracies, 96.0% (image-wise) and 96.1% (object-wise), with slightly increased computation time due to region proposal network (120 ms per image).

## 2.2 PROPOSED METHODS FOR ROBOTIC GRASPS

### 2.2.1 Problem Description

The goal of the problem is to predict 5D robotic grasp representations [10, 16] for multiple objects from a given color image (RGB) and possibly depth image (RGB-D) where a 5D robotic grasp representation consists of location  $(x, y)$ , orientation  $\theta$ , gripper opening width  $w$ , and parallel gripper plate size  $h$ , as illustrated in Fig. 2.2 (a). Then, the 5D robotic grasp representation

$$\{x, y, \theta, w, h\}$$

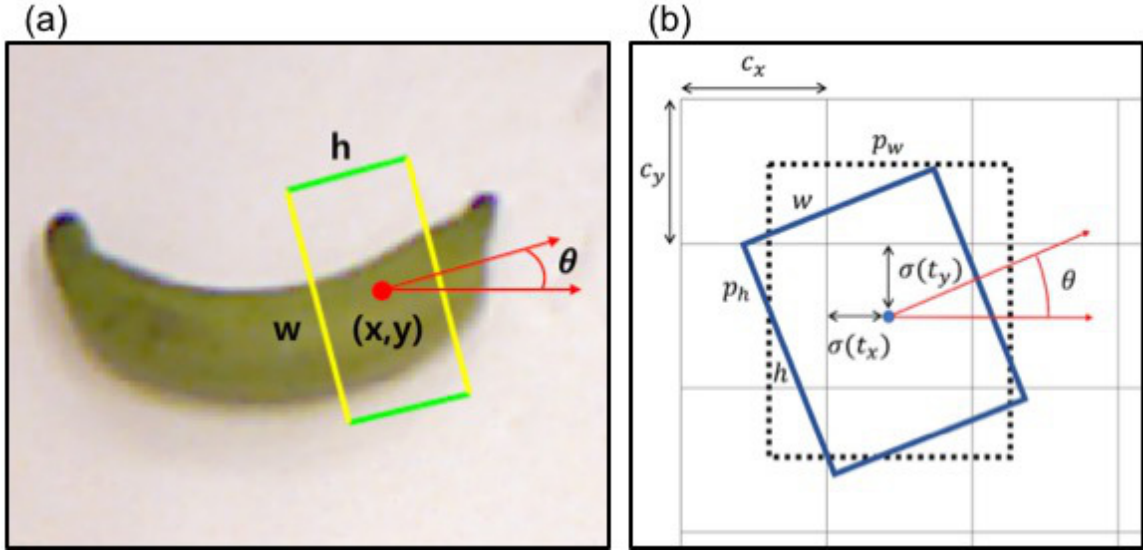


Figure 2.2: (a) A 5D grasp representation with location  $(x, y)$ , orientation  $\theta$ , gripper opening width  $w$  and plate size  $h$ . (b) For the  $(2, 2)$  grid cell, all parameters for 5D grasp representation are illustrated including a pre-defined anchor box (black dotted box), a 5D grasp representation (blue box).

in camera based vision coordinate system should be transformed into a new 5D grasp representation  $\{\tilde{x}, \tilde{y}, \tilde{\theta}, \tilde{w}, \tilde{h}\}$  in actual robot coordinate system so that they can be used for actual robot grasping task.

## 2.2.2 Reparametrization of 5D Grasp Representation and Grasp Probability

MultiGrasp estimates 5D grasp representation  $\{x, y, \theta, w, h\}$  as well as grasp probability (confidence)  $z$  for each grid cell by reparameterizing  $\theta$  to be  $c = \cos \theta$ ,  $s = \sin \theta$  [17]. In other words, 7 parameters  $\{x, y, c, s, w, h, z\}$  are directly estimated using deep learning based regressors in MultiGrasp. This approach has also been used in YOLO, object detection deep network [37]. Inspired by YOLO9000, a better and faster deep network for object detection than YOLO [14], we propose the following reparametrization of 5D grasp representation and grasp probability for robotic grasp detection as follows:

$$\{t^x, t^y, \theta, t^w, t^h, t^z\}$$

where  $x = \sigma(t^x) + c_x$ ,  $y = \sigma(t^y) + c_y$ ,  $w = p_w \exp(t^w)$ ,  $h = p_h \exp(t^h)$ , and  $z = \sigma(t^z)$ . Note that  $\sigma(\cdot)$  is a sigmoid function,  $\exp(\cdot)$  is an exponential function,  $p_h$ ,  $p_w$  are the pre-defined height and width of an anchor box, respectively, and  $(c_x, c_y)$  are the location of the top left corner of each grid cell (known). Thus, deep neural network for robotic grasp detection of our

proposed methods will estimate  $\{t^x, t^y, \theta, t^w, t^h, t^z\}$  instead of  $\{x, y, \theta, w, h, z\}$ . These parameters are illustrated in Fig. 2.2 (b). Note that  $x, y, w, h$  are properly normalized so that the size of each grid cell is  $1 \times 1$ . Lastly, the angle  $\theta$  will be modeled as a discrete value instead of a continuous value, which is different from MultiGrasp. This discretization of the angle in robotic grasp detection was also used in [36].

**(x, y) coordinates in each grid cell (offset).** Instead of predicting  $(x, y)$  in the image coordinate, our proposed methods will predicting the location of robotic grasp by estimating the  $(x, y)$  offset from the top left corner of each grid cell  $(c_x, c_y)$ . For  $S \times S$  grid cells,

$$(c_x, c_y) \in \{(c_x, c_y) | c_x, c_y \in \{0, 1, \dots, S - 1\}\}$$

Thus, for a given  $(c_x, c_y)$ , the range of  $(x, y)$  will be

$$c_x < x < c_x + 1, \quad c_y < y < c_y + 1$$

due to the re-parametrization using sigmoid functions.

**w, h coordinates in each cell (anchor box).** Anchor box approach has also been useful for object detection [14], so we adopt it to our robotic grasp detection. Due to the re-parametrization using anchor box, estimating  $w, h$  is converted into estimating  $t^w, t^h$ , which are related to the expected values of various sizes of  $w, h$ , and then classifying the best grasp representation among all anchor box candidates. In other words, this re-parametrization changes regression problems for  $w, h$  into regression + classification problems. We propose to use the following 7 anchor boxes:

$$(p_w, p_h) \in \{(0.76, 1.99), (0.76, 3.20), \\ (1.99, 0.76), (1.99, 1.99), (1.99, 3.20), \\ (3.20, 3.20), (3.20, 0.76)\}.$$

### 2.2.3 Loss Function for Robotic Grasp Detection

We proposed a novel loss function for robotic grasp detection considering the following items.

**Angle in each cell (discretization).** MultiGrasp re-parameterized the angle  $\theta$  with  $c = \cos \theta$  and  $s = \sin \theta$  so that estimating  $c, s$  yields the estimated  $\theta = \arctan(s/c)$ . Thus, MultiGrasp took regression approach for  $\theta$ . We proposed to convert this regression problem for estimating  $\theta$  into the classification problem for  $\theta$  among finite number of angle candidates in



$[0, \pi]$ . Specifically, we model that  $\theta \in \{0, \pi/18, \dots, \pi\}$ . Along with data augmentation for different angles every epoch, we were able to observe substantial performance improvement. Similar angle discretization for robotic grasp detection was also used in [36].

**Grasp probability (new ground truth).** Predicting grasp probability is crucial for multi-box approaches such as MultiGrasp. Conventional ground truth for grasp probability was 1 (graspable) or 0 (not graspable) as used in [17]. Inspired by YOLO9000, we proposed to use IOU (Intersection Over Union, Jaccard index) as the ground truth for grasp probability: the ground truth for grasp probability is

$$z^g = \frac{|P \cap G|}{|P \cup G|} \quad (\text{II.1})$$

where  $P$  is the predicted grasp rectangle,  $G$  is the ground truth grasp rectangle, and  $|\cdot|$  is the area of the inner set.

**Proposed loss function.** We propose to use the follow cost function to train robotic grasp detection networks that we will describe in the next subsection: For the output vector of the deep neural network  $(t^x, t^y, \theta, t^w, t^h, t^z)$  and the ground truth  $\{x^g, y^g, \theta^g, w^g, h^g, z^g\}$ ,

$$\begin{aligned} L(t^x, t^y, \theta, t^w, t^h, t^z) = & \\ & \lambda_{\text{coord}} \sum_{i=1}^{S^2} \sum_{j=1}^A m_{ij}^{\text{obj}} [(x_i^g - x_i)^2 + (y_i^g - y_i)^2] + \\ & \lambda_{\text{coord}} \sum_{i=1}^{S^2} \sum_{j=1}^A m_{ij}^{\text{obj}} [(w_{ij}^g - w_{ij})^2 + (h_{ij}^g - h_{ij})^2] + \\ & \lambda_{\text{prob}} \sum_{i=1}^{S^2} \sum_{j=1}^A m_{ij}^{\text{obj}} [(z_i^g - z_i)^2] + \\ & \lambda_{\text{class}} \sum_{i=1}^{S^2} \sum_{j=1}^A m_{ij}^{\text{obj}} \text{CrossEntropy}(\theta_i^g, \theta_i) \end{aligned}$$

where  $x_i, y_i, w_{ij}, h_{ij}, z_i$  are functions of  $(t^x, t^y, t^w, t^h, t^z)$ , respectively,  $S^2$  is the number of grid cells and  $A$  is the number of anchor boxes (7 in our case). We set  $\lambda_{\text{coord}} = 1$ ,  $\lambda_{\text{prob}} = 5$  and  $\lambda_{\text{class}} = 1$ . We set  $m_{ij} = 1$  if the ground truth  $(x^g, y^g)$  is in the  $i$ th cell and  $m_{ij} = 0$  otherwise.

## 2.2.4 Proposed FCNN Architecture

We chose three well-known deep neural networks for image classification tasks Alexnet [11] (base network for MultiGrasp [17]), Darknet-19 (similar to VGG-16 [40] that was used in [18],

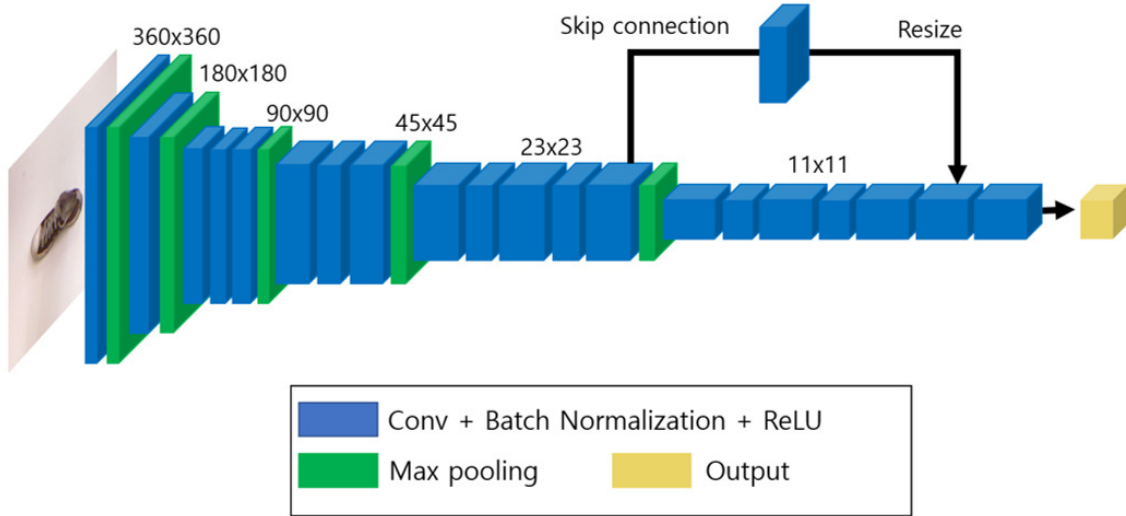


Figure 2.3: Proposed FCNN architecture based on Darknet-19.

but with much smaller memory requirement for similar performance) [14], and Resnet-50 [12] (base network for [18, 36]). These pre-trained networks were modified to yield robotic grasp parameters and their fully connected (FC) layers were replaced by  $1 \times 1$  convolution layers to make FCNN architecture so that images with any size (*e.g.*, high resolution images) can be processed. Most previous robotic grasp detection methods use  $227 \times 227$  resized images as input, but our proposed FCNN based methods can process higher resolution images. We chose to process  $360 \times 360$  images for grasp detection without resizing. Skin connection layer was also added so that fine grain features can be used. For example, a passthrough layer was added in between the final  $3 \times 3 \times 512$  layer and the second to last convolutional layer for Darknet-19 as illustrated in Fig. 2.3 [14]. Similarly, we added similar skip connection for Resnet-50 in between the convolutional layer right before the last max pooling layer and detection layer. Unfortunately, we did not add skip connection for Alexnet since the pre-trained network did not provide access to inner layers.

### 2.2.5 Learning-based Vision-Robot Calibration

For a successful robot grasping, accurately predicted 5D grasp representation  $\{x, y, \theta, w, h\}$  in vision coordinate system must be converted into 5D grasp representation  $\{\tilde{x}, \tilde{y}, \tilde{\theta}, \tilde{w}, \tilde{h}\}$  in actual robot coordinate system considering gripper configuration. Thus, accurate calibration between vision and robot coordinate systems is critical for robotic grasping. Our robot is equipped with a gripper whose maximum open distance  $w$  is 27.5 mm. In order to grasp small objects whose widths are 10-20 mm, the calibration error between vision and robot coordinates should be less than or equal to 1-2 mm.

We proposed a learning-based, fully automatic vision-robot calibration method as illustrated in Fig. 2.4: (1) a small known object (round shape in our case) is placed in a known location, (2) the robot moves the object to a random location, (3) the robot places the object, (4) the robot is away from field of view, (5) vision system predicts 5D grasp representation, and (6) the procedure is repeated to collect many samples. Then, 5D grasp representations in both vision coordinate and robot coordinate can be mapped using linear or nonlinear regressions or using simple nonlinear neural networks. For simplicity, we calibrated only  $x, y$  with affine transformation using LASSO [41] assuming known  $w$  (maximum open width of the gripper), known  $h$  (fixed gripper), and relatively good tolerance for  $\theta$ . The ranges of  $x, y$  in our robot coordinate are 150 to 326 mm, -150 to 150 mm, respectively, and the ranges of  $x, y$  in our vision coordinate are 160 to 290 pixel, 50 to 315 pixel, respectively. One pixel corresponds to about  $1.35 \times 1.13 \text{ mm}^2$ .

Fig. 2.5 shows that calibration error (in mm) is in general decreasing as the number of samples is increasing and the error is below 1.5 mm which is close to one pixel in vision if there are more than 40 samples. Note that since there are 6 LASSO coefficients for mapping  $x, y$ 's, theoretically only 3 points should be enough to determine all 6 coefficients. However, in practice, much more samples are necessary to ensure good calibration accuracy. This result

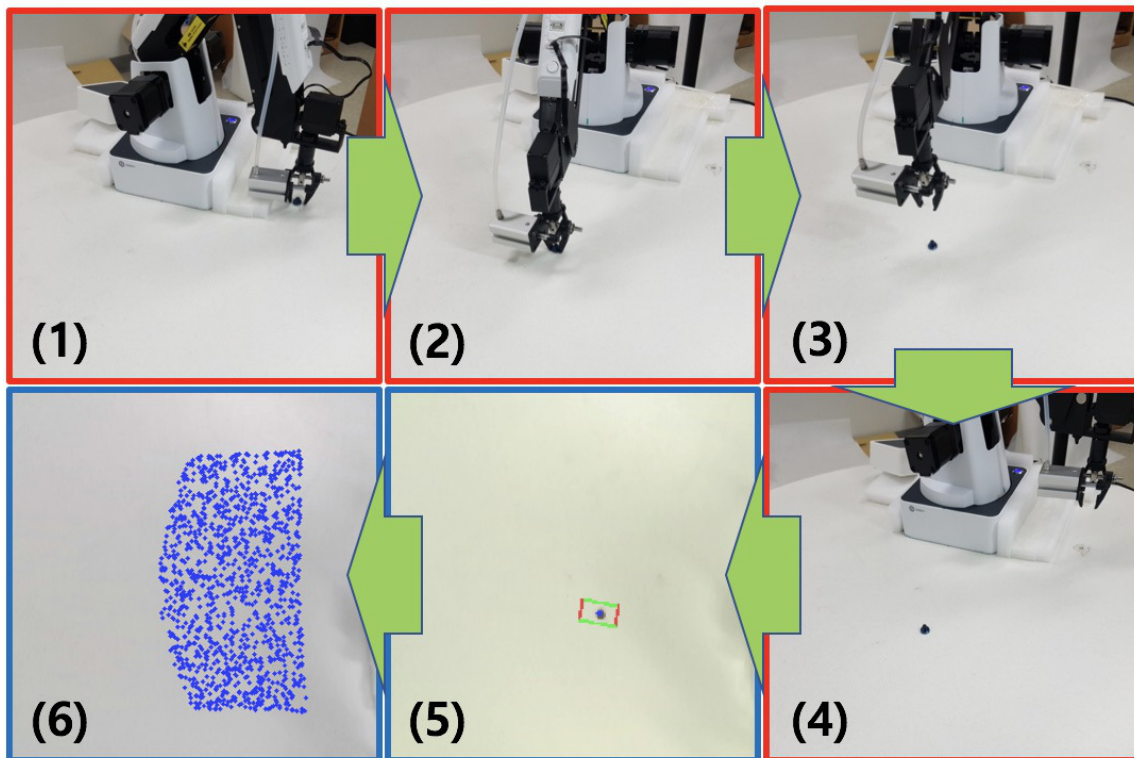


Figure 2.4: Proposed learning-based vision-robot calibration.

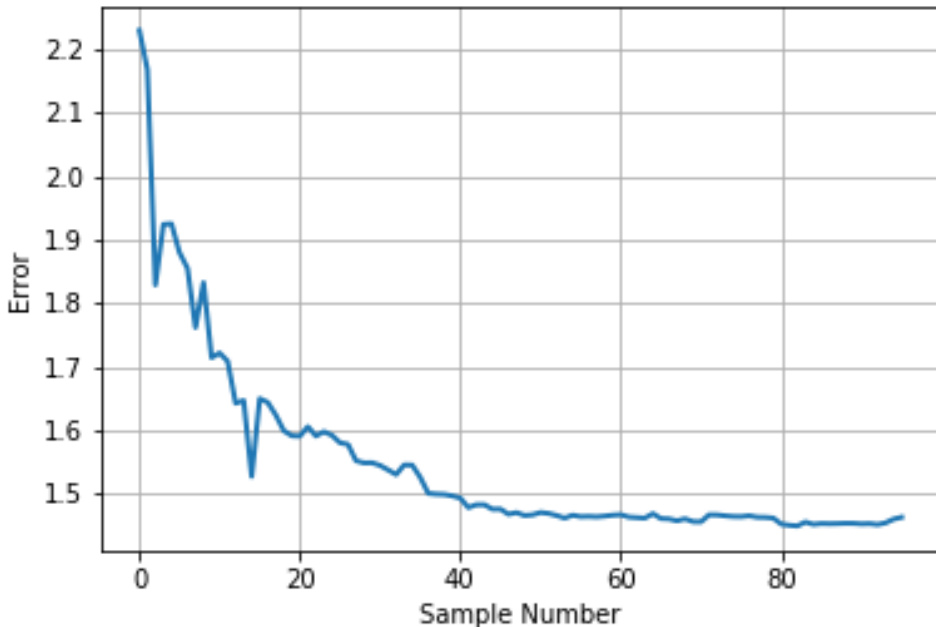


Figure 2.5: Calibration error (in mm) for  $x, y$  in robot coordinate system over increasing number of learning samples.

implies that using high resolution images seem important for successful grasping due to potential high accuracy of calibration.

## 2.3 EXPERIMENTS AND EVALUATION

### 2.3.1 Evaluation with Cornell Dataset

We performed benchmarks using the Cornell grasp detection dataset [15, 16] as shown in Fig. 2.6. This dataset consists of 855 images (RGB color and depth) of 240 different objects with the ground truth labels of a few graspable rectangles and a few not-graspable rectangles. Note that we cropped images with  $360 \times 360$ , but did not resize it to  $224 \times 224$ . Five-fold cross validation was performed and average prediction accuracy was reported for image-wise and object-wise splits. When the difference between the output orientation  $\theta$  and the ground truth orientation  $\theta^g$  is less than 30 degree, then IOU or Jaccard index in Eq. (II.1) that is larger than a certain threshold (*e.g.*, 0.25, 0.3) will be considered as a success grasp detection.

The same metric for accuracy has been used in other previous works [16, 17, 34].

All proposed methods were implemented using pyTorch and trained with 500 epochs and data augmentation that took about 4 hours of training. For fair comparison, we implemented the

work of Lenz *et al.* [15,16] and MultiGrasp [17] using MATLAB or Tenforflow. They achieved similar performance and computation time that were reported in their original papers. All algorithms were tested on the platform with a single GPU (NVIDIA GeForce GTX1080Ti), a single CPU (Intel i7-7700K 4.20GHz) and 32GB memory.

### 2.3.2 Evaluation with 4-axis Robot Arm and RGB-D

We also evaluated our proposed methods with a small 4-axis robot arm (Dobot Magician, Shenzhen YueJiang Tech Co., Ltd, China, Fig. 1.2 (Right)) and a RGB-D camera (Intel RealSense D435, Intel, USA) attached to have the field-of-view including the robot and its workspace from the top. The following 6 novel objects (toothbrush, candy, earphone cap, cable, styrofoam bowl, L-wrench were used for real grasp tasks as shown in Fig. 2.7. After our learning-based vision-robot calibration, for each object, 5 repetition were performed. If the robot arm is holding an object for more than 3 sec, it is counted as a success grasp.

## 2.4 RESULTS

### 2.4.1 Evaluation Results on Cornell Dataset

Table 2.1 summarizes all evaluation results on the Cornell robotic grasp dataset for all our proposed methods. Our proposed methods yielded state-of-the-art performance, up to 96.6% prediction accuracy for image-wise split with any metric with state-of-the-art computation time of 3-20 ms. For object-wise split, our proposed methods yielded comparable results for less tolerant metrics (25%, 30%), but yielded state-of-the-art performance for more strict metrics



Figure 2.6: Images from Cornell grasp detection dataset.



Figure 2.7: Novel objects for real robot grasping tasks.

Table 2.1: Performance summary on the Cornell dataset with IOU metric. Our proposed methods yielded state-of-the-art prediction accuracy in both image-wise and object-wise splits with state-of-the-art computation time. Note that Resnet-50, Darknet-19, Alexnet require 82.6, 48.5, and 6.0MB memory, respectively. Performance unit is in % unless specified.

Size	Offset	Deep network	Data type	Image-wise				Object-wise			
				25%	30%	35%	40%	25%	30%	35%	40%
360	O	Resnet-50	RG-D	<b>96.6</b>	94.6	91.5	<b>86.7</b>	95.4	92.5	<b>88.5</b>	82.5
360	O	Resnet-50	RGB	<b>96.6</b>	93.7	91.0	<b>85.7</b>	95.1	92.5	<b>88.7</b>	<b>82.9</b>
360	O	Darknet-19	RG-D	<b>96.6</b>	<b>95.4</b>	<b>92.4</b>	<b>87.4</b>	94.7	92.0	<b>89.0</b>	<b>83.2</b>
360	O	Darknet-19	RGB	<b>96.4</b>	93.6	90.7	<b>86.5</b>	94.0	91.3	86.5	80.3
360	-	Darknet-19	RGB	89.8	87.6	84.9	80.1	87.7	85.4	81.6	72.5
224	O	Darknet-19	RGB	93.5	89.7	85.4	77.7	91.5	88.0	81.9	75.6
360	O	Alexnet	RGB	93.6	90.3	86.5	80.2	91.1	86.8	81.0	73.5
224	-	Alexnet	RGB	89.1	79.5	69.0	57.3	86.7	76.6	64.6	51.1
227		Chu [18]		96.0	94.9	92.1	84.7	<b>96.1</b>	<b>92.7</b>	87.6	82.6
227		Guo [36]#a		93.2	91.0	85.3	-	82.8	79.3	74.1	-
227		Guo [36]#c		86.4	83.6	76.8	-	89.1	85.1	80.5	-
227		Kumra [34]		89.2	-	-	-	88.9	-	-	-
227		Redmon [17]		88.0	-	-	-	87.1	-	-	-
227		Lenz [15]		73.9	-	-	-	75.6	-	-	-
227		Jiang [10]		60.5	-	-	-	58.3	-	-	-

(35%, 40%), demonstrating that our methods yielded highly accurate grasp detection information with true real-time computation. The results of Table 2.1 also indicate the importance of good deep network (Darknet, Resnet over Alexnet), of using re-parametrization (Offset), and of using high resolution images as input for better performance. Fig. 2.8 qualitatively illustrates some of these points. Using low resolution image and/or simple network architecture seems to result in missing small graspable candidates as indicated with missing small graspable areas around shoe neck.

#### 2.4.2 Evaluation Results with 4-Axis Robot Arm

Fig. 2.9 illustrates our robot grasp experiment with “candy” object. While previous methods or our method with low image resolution tend to grasp candy part, our proposed method yielded grasp areas around stick part of the candy and our robot actually grasped it as shown in the figure. Table 2.2 summarizes our robot experiments showing that our proposed method with

high resolution yielded 90% grasp success rate while other methods yielded 53% or less.

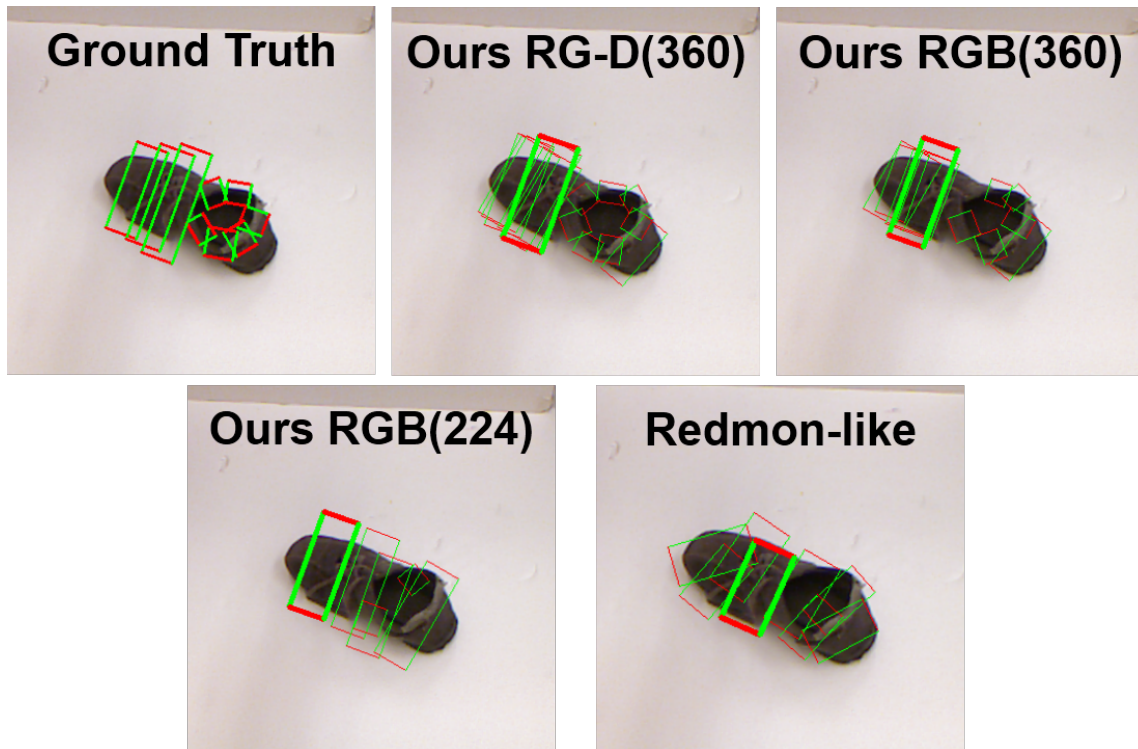


Figure 2.8: One grasp detection results with different image resolution, data type, and with different deep network. All methods were able to detect large grasp areas, but the methods with small deep network and/or low image resolution missed some small grasp areas.

Table 2.2: Performance summary of real robotic grasping for 6 novel, small objects with 5 repetitions. For Lenz and Redmon, our in-house implementations (modifications) were used after validating their performance with the Cornell dataset. Darknet implementation was used for Ours with resized image (224) and with high resolution image (360).

Object	Lenz*	Redmon*	Ours(224)	<b>Ours(360)</b>
toothbrush	80%	80%	60%	<b>100%</b>
candy	0%	60%	20%	<b>100%</b>
earphone cap	40%	20%	<b>80%</b>	<b>80%</b>
cable	0%	0%	40%	<b>100%</b>
styrofoam bowl	0%	20%	<b>80%</b>	60%
L-wrench	80%	100%	40%	<b>100%</b>
Average	33%	47%	53%	<b>90%</b>

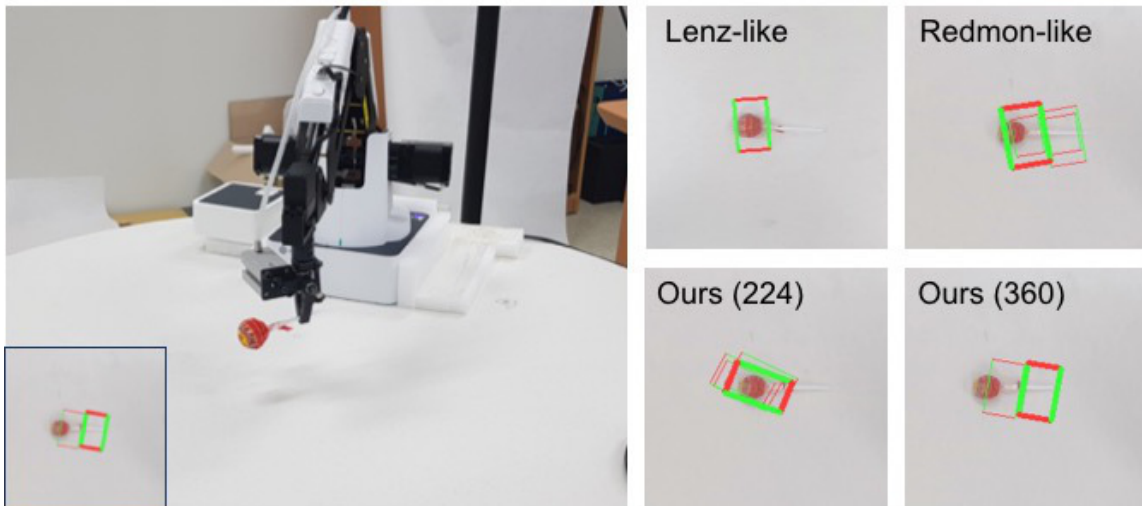


Figure 2.9: An illustration of our robot grasp experiment with “candy” (Left) and multigrasp detection results for “candy” using 4 different methods. Ours (360) successfully detect stick part of the candy.



---

# Efficient Module Based Single Image Super Resolution for Multiple Problems

---

## 3.1 Background and Related Works

**Deep learning based super resolution.** Dong *et al.* used convolutional neural network (CNN) for SR problem (SRCNN) and achieved significant improvement in performance over other conventional non-deep learning based methods [4]. An LR image is upscaled using bicubic interpolation and then CNN was applied to restore HR details. Soon after, Kim *et al.* proposed a deep neural network using residual learning (VDSR) and showed improved PSNR performance over SRCNN [5]. In this method, CNN was trained not to yield a HR image, but a residual image for the difference between an interpolated LR image and the ground truth HR image. VDSR also used a deeper CNN network than SRCNN.

Lai *et al.* proposed a Laplacian pyramid super resolution network (LapSRN) that combines multiple models and uses progressive reconstruction from  $\times 8$  to  $\times 4$  to  $\times 2$  to HR ( $\times 1$ ) [26].

residual blocks [29] to significantly increase the size of the receptive field and to include local context information so that state-of-the-art performance for  $\times 4$  SR problem can be obtained in terms of PSNR and SSIM [3]. SRGAN was also proposed with the same network structure as SRResNet, but with different training based on a discriminator network. SRGAN yielded

visually pleasing outputs while PSNR of SRGAN was lower than that of SRResNet since SRResNet yielded an average of many possible outputs while SRGAN yielded one of many possible outputs.

Recently, Lim *et al.* won the NTIRE 2017 challenge [25] for SR problems using so-called EDSR (Enhanced Deep Super-Resolution network) that enhanced SRResNet by eliminating batch normalization and by stacking deeper layers (residual blocks from 16 to 32, filter channels from 64 to 256) [1]. EDSR also used L1 loss instead of L2 loss for better PSNR. NTIRE 2017 consisted of two Tracks for known (bicubic) and unknown blurs and for each Track, there were three different downsampling rates ( $\times 2$ ,  $\times 3$ ,  $\times 4$ ). EDSR won the 1st place for NTIRE 2017 by outperforming SRResNet for all public dataset including DIV2K, NTIRE 2017's new dataset [1].

**Deep learning based denoising and deblurring.** Patch based denoising methods yielded superior denoising results compared to conventional denoising techniques [42], but they are usually slow in computation and have so called rare patch issue so that these are less effective for unique patterns in an image. Recently, there have been several attempts to outperform patch based denoisers such as BM3D using deep learning based approaches. Jain and Seung demonstrated that denoising is possible using CNN [43]. Burger *et al.* proposed a multi layer perceptron based denoiser and showed that it is challenging, but possible to obtain good denoising performance over conventional state-of-the-art methods such as BM3D [44]. Xie *et al.* proposed a deep network for denoising and inpainting [45]. Recently, Lefkimiatis investigated a combined method of conventional non-local patch based denoiser and deep learning based denoiser [46]. Zhang *et al.* proposed a so-called DnCNN with multiple CNN blocks (similar to VDSR) to yield a residual (Gaussian noise) and to yield superior performance to other denoisers including BM3D [2].

In particular, DnCNN has greatly improved the performance of denoising and deblurring tasks with a simple deep convolution layer and residual learning.

## 3.2 Method

### 3.2.1 Modular Approach

We decomposed the original problems in NTIRE 2018 Tracks 1, 2, 3 into subproblems as illustrated in Figures 1.4 (a) (Track 1) and 1.4 (b) (Tracks 2, 3) and efficiently recycled trained deep neural networks for a number of subproblems. Figure 3.1 illustrates our detailed network architectures for all problems in Tracks 1, 2, 3, called efficient module based super resolution

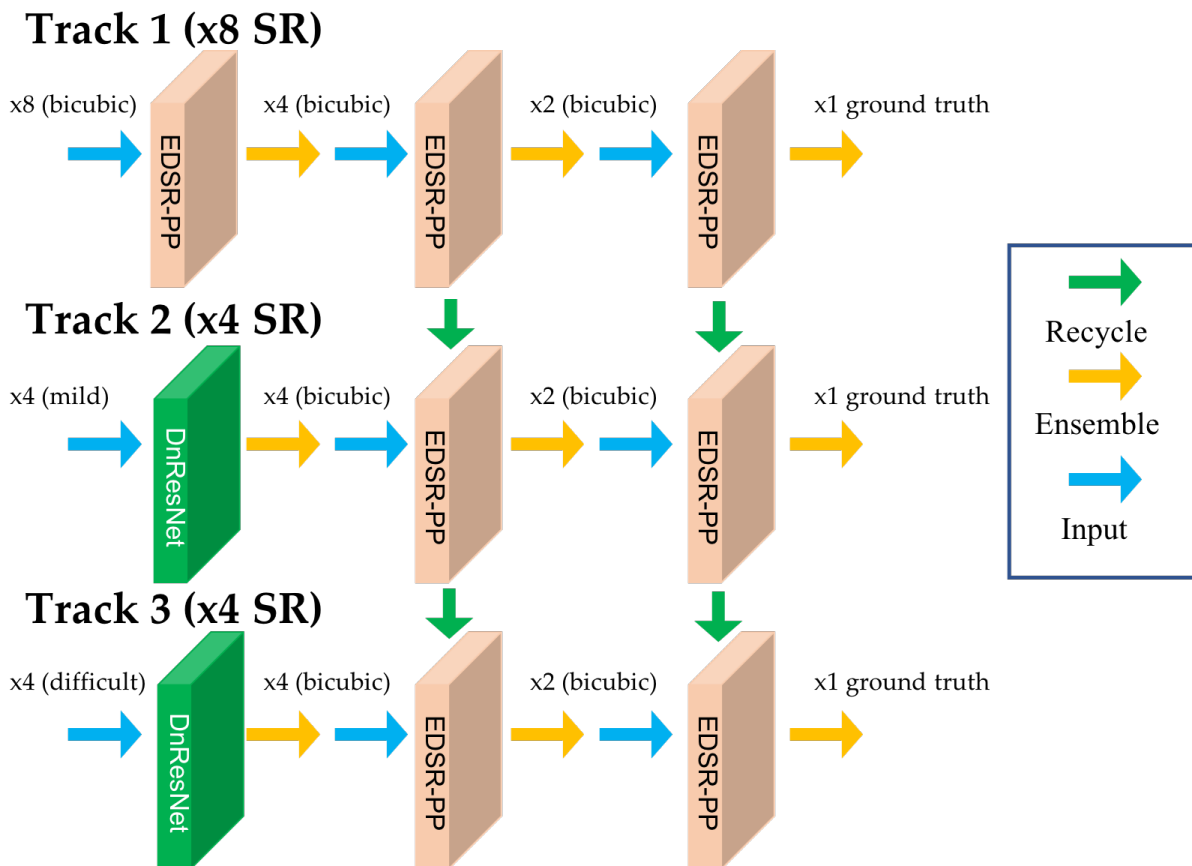


Figure 3.1: Modular approach for multiple SR problems. Among 9 modules, 5 modules required long training while 4 modules can be recycled with short fine tuning.

(EMBSR) network. This modular approach allows us to train networks module-by-module and to efficiently recycle trained modules for multiple SR problems (see Figure 3.1 to see that among 9 modules, only 5 modules require long training, while 4 modules can recycle already trained networks with relatively short fine tuning). This modular architecture also yielded effectively deeper networks with more feature map channels when limited computation and memory resource are available. Each module can generate ensemble output for each subproblem to increase the PSNR performance without increasing the complexity of networks. Lastly, modular approach allowed us to separate SR subproblems from the problem of denoising and deblurring for Tracks 2, 3. Due to this separation, significant performance improvement was achieved by utilizing optimal deep networks for different problems (*e.g.*, EDSR for SR problem and DnCNN for denoising/deblurring problem) and by aligning an input image and an intermediate target image ( $\times 4$  bicubic downsampled image) for training denoiser/deblur networks.

Our EMBSR network for Track 1 ( $\times 8$  bicubic) consists of three EDSR-PP networks as illustrated in the top of Figure 3.1. For training each module network, we downsampled ground truth images using bicubic downsampling to generate target images for each module ( $\times 2$  bicubic downsampled images,  $\times 4$  bicubic downsampled images). Then, all EDSR-PP modules were trained with given input  $\times 8$  bicubic downsampled images and generated  $\times 4$  bicubic downsampled images, input  $\times 4$  bicubic downsampled images and generated  $\times 2$  bicubic downsampled images, and  $\times 2$  bicubic downsampled images and ground truth images. A solution for Track 1 ( $\times 8$  single image SR) was created by concatenating three trained modules. Note that ensemble output is possible by having 8 variants of an input image (4 rotations  $\times$  2 left-right flips) for each neural network module. This procedure substantially improved performance. Further fine tuning is also possible. Each module is trained with perfect bicubic downsampled input images, but the ensemble output of each module contains errors from them. In EBMSR for Track 1, the second EDSR-PP module can be re-trained using ensemble output images of the first EDSR-PP module and then the third EDSR-PP module can be re-trained using ensemble output images of the re-trained second EDSR-PP module, sequentially. In our simulations, training each EDSR-PP module took about 3 days for 300 epochs and re-training each module took about 1 day for 100 epochs. Our EMBSR network for Tracks 2, 3 is similar to the EMBSR network for Track 1, but with replacing the first EDSR-PP module with DnResNet module, as illustrated in the middle and bottom of Figure 3.1, respectively. The second and third “trained” EDSR-PP modules for Track 1 can be recycled in Tracks 2, 3 as shown in Figure 3.1 (green arrows). The first DnResNet module for tackling Track 2 can be trained using given input training data and target  $\times 4$  bicubic downsampled images. Image registration between input and target images using a translation motion was critical to significantly improve the performance of

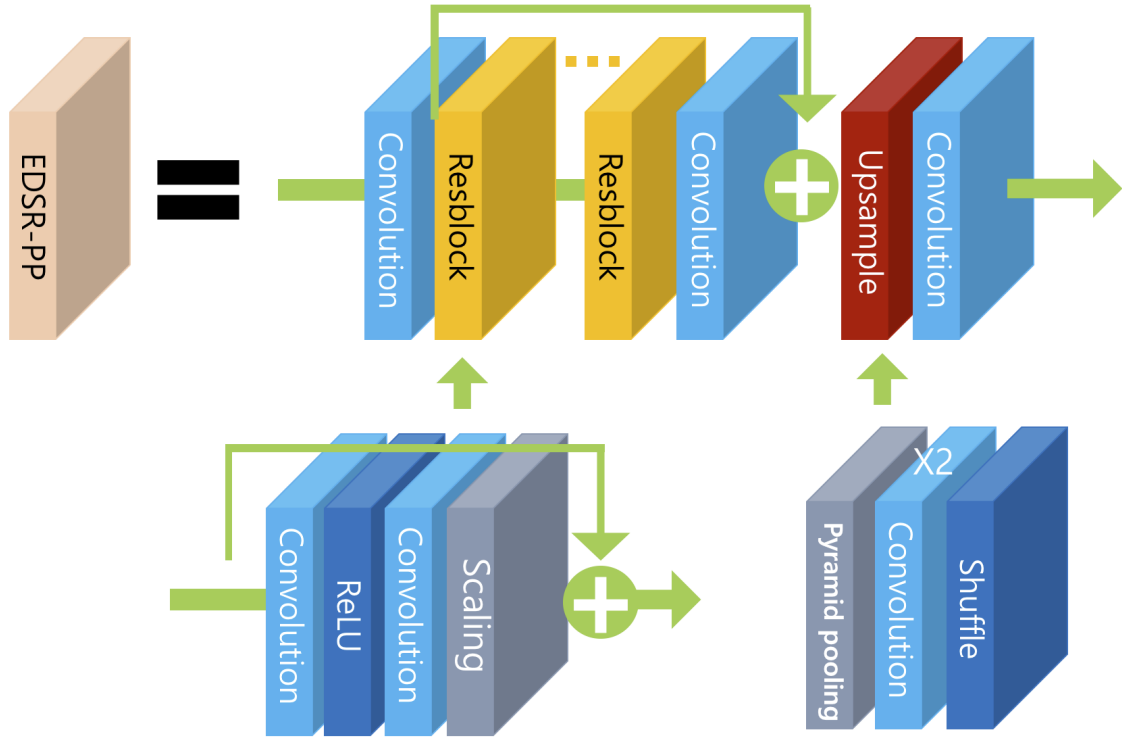


Figure 3.2: An illustration of our proposed EDSR-PP. Upsampling lay of the original EDSR [1] was replaced with pyramid pooling structure.

DnResNet as well as baseline DnCNN. For Track 3, similar approach can be applied. Then, solutions for Tracks 2, 3 can be obtained by concatenating trained DnResNet and two other trained EDSR-PP networks. Further improvement was achieved by sequentially re-training the second EDSR-PP module using ensemble output images of the first DnResNet module, and then fine tuning the third EDSR-PP module using ensemble output images of the re-trained second EDSR-PP module for both Tracks 2 and 3.

### 3.2.2 SR Module: EDSR-PP (Pyramid Pooling)

We propose a new SR network, EDSR-PP, based on a state-of-the-art SR network, EDSR [1]. EDSR-PP incorporates pyramidal pooling [28] into the upsampling layer of the original EDSR as illustrated in Figure 3.2.

The number of residual blocks in EDSR-PP was 32 and the same network architecture was used for Typically, the receptive field size of deep learning based image processing corresponds to how much context information is included. The deeper the CNN network is, the larger the receptive field size is. However, in CNN based deep networks for image processing, this receptive field size may not be large enough to receive global context information. Pyramid pooling [28]

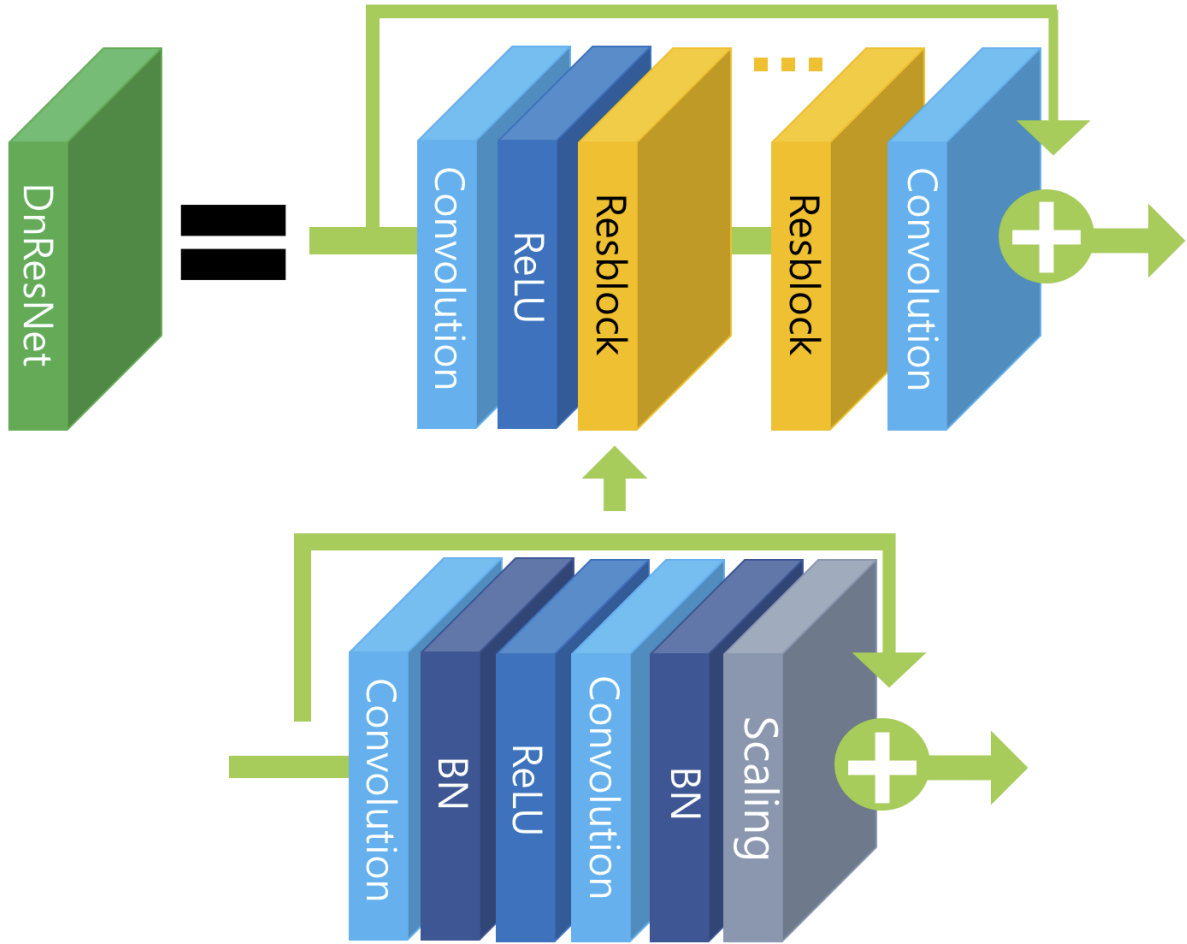


Figure 3.3: An illustration of our proposed DnResNet. Unlike DnCNN that uses CNN layers [2], residual blocks (Resblock) were used as a basic building block.

is a recent method to resolve this issue so that both local and global context information can be utilized for image segmentation problems. We incorporated it into EDSR for SR problem. In contrast to the up-sampling layer of EDSR, pyramid pooling firstly executes average pooling and performs convolution for each of the four pyramid scales. Then, these are concatenated in the existing feature map. This process allows both local and global context information to be utilized. Four pyramid scales were used in our EDSR-PP with  $1 \times 1$ ,  $2 \times 2$ ,  $3 \times 3$ , and  $4 \times 4$  and our proposed EDSR-PP yielded better performance than EDSR.

### 3.2.3 Denoising / Deblurring Module: DnResNet

We also propose a novel denoising / deblurring network, DnResNet, based on one of the state-of-the-art methods, DnCNN [2] for denoising / deblurring problem. DnCNN uses residual learning (skip connection between input and output) and multiple convolution blocks with con-

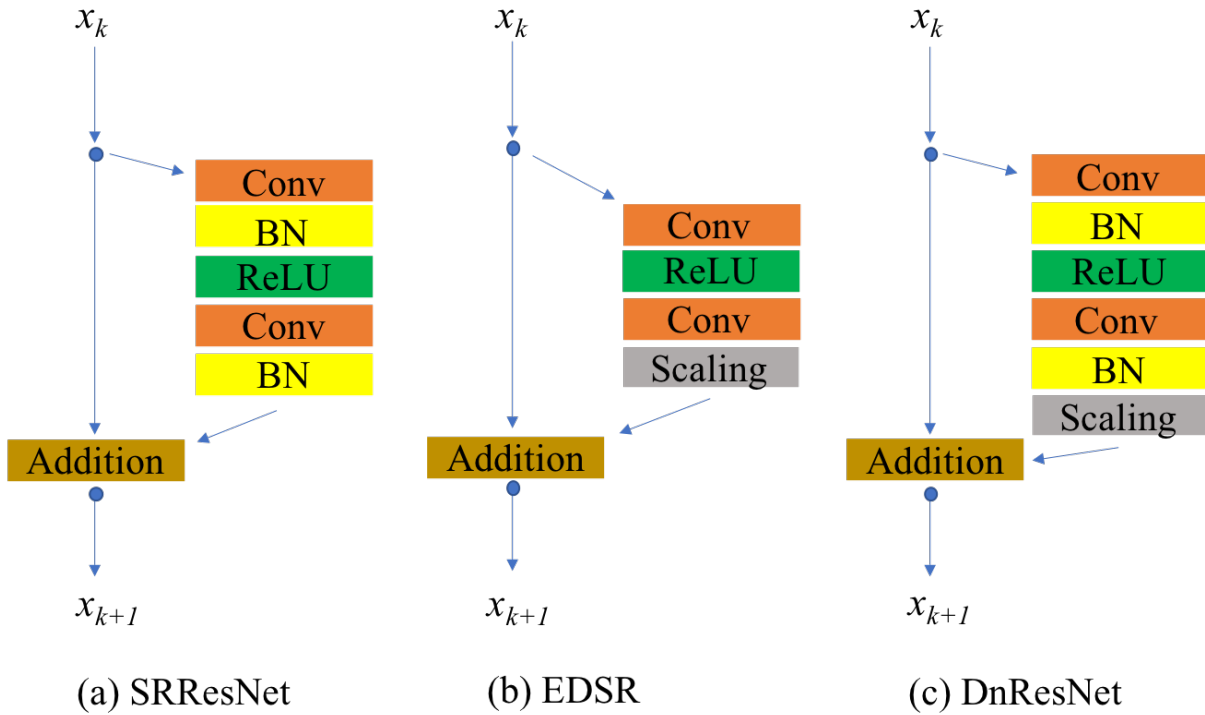


Figure 3.4: Comparison of residual blocks for SRResNet [3], EDSR [1], and our DnResNet.

volution - batch normalization - ReLU layers. Our DnResNet simply replaces all convolution blocks with our residual blocks as shown in Figure 3.3. Using residual blocks further increased receptive fields efficiently without concatenating more deep convolution layers. DnCNN used 64 feature map channels while our DnResNet used 128 feature map channels. For residual blocks, EDSR removed batch normalization layers from and added 0.1 scaling to the residual block of SRResNet as shown in Figure 3.4 for improved performance and numerical stability of training in SR problem. However, we found that it is advantageous to keep batch normalization layers for denoising and deblurring problems. So, we modified the residual block of EDSR by adding two batch normalization layers again. Note that our residual block is equivalent to the original residual block of SRResNet except for 0.1 residual scaling. Note also that our proposed DnResNet utilized similar residual blocks as SRResNet, but overall network architectures are quite different. Our proposed DnResNet with residual blocks outperformed DnCNN with convolution blocks for denoising and deblurring problems.

## 3.3 Experiment

### 3.3.1 Dataset

The DIV2K dataset from the NTIRE 2018 challenge was used in all simulations of this article. DIV2K is a high quality (2K resolution) image data set from the NTIRE 2017 challenge [25]. For the same ground truth HR images,  $\times 8$  bicubic downsampled images were provided for Track 1,  $\times 4$  downsampled images with unknown blur kernels and mild noise were provided for Track 2, and  $\times 4$  downsampled images with unknown, difficult blur kernels and noise were provided for Track 3. For each track, 800 training images, 100 validation images, and 100 test images were given. In this article, we only use 10 images (801 to 810).

### 3.3.2 Training and Alignment

Training procedures are well described in Section ???. Mini batch size was 16 and patch size was  $48 \times 48$ . For individual module training, 300 epochs were run with learning rates of  $10^{-4}$  for 1 to 100 epochs and  $10^{-5}$  for 101 to 300 epochs. It took about 3 days to run 300 epochs for each module network. Re-training learning rate was set to  $10^{-5}$  for 100 epochs.

We found that given input images of Tracks 2 and 3 and  $\times 4$  bicubic downsampled ground truth images are not well aligned. In principle, these misalignment should be taken care of by deep neural networks during training. However, aligning input and target images as much as possible helped to achieve improved performance. Given input images of Tracks 2 / 3 and  $\times 4$  bicubic downsampled ground truth images were aligned using image intensity based image registration tool in MATLAB with translation motion only. Bicubic interpolation was used for sub-pixel accuracy.

### 3.3.3 DIV2K Validation Set Results

Table 3.1 shows performance results for DIV2K validation set, comparing various SR methods such as bicubic interpolation, SRCNN [4], VDSR [5], EDSR [1] and our proposed EMBSR. Our EDSR-PP based EMBSR method yielded improved PSNR results for SR problems with different scales ( $\times 2$ ,  $\times 4$ , and  $\times 8$ ) over other methods. This result demonstrated that our proposed SR module, EDSR-PP, yielded state-of-the-art SR performance.

Table 3.2 showed that our proposed DnResNet outperformed current state-of-the-art denoising / deblurring method, DnCNN [2], with both misaligned and aligned data set. It seems that aligning given input and target images was critical to achieve high performance in denoising and deblurring. imized for train data (2k resolution), so it gets low results in other datasets.



Table 3.1: PSNR (dB) results of different methods for DIV2K validation data set: SRCNN [4], VDSR [5], EDSR [1], and our proposed EMBSR.

	Bicubic	SRCNN	VDSR	EDSR	<b>EMBSR</b>
$\times 2$	31.01	33.05	33.66	35.12	<b>35.87</b>
$\times 4$	26.66	27.70	28.17	29.38	<b>29.89</b>
$\times 8$	24.51	-	-	26.00	<b>26.22</b>

Table 3.2: Performance comparison between architectures on the DIV2K validation set (PSNR in dB).

DnCNN [2]	DnResNet	DnCNN [2] (aligned)	DnResNet (aligned)
21.005	25.359	29.439	<b>30.281</b>

Trained EDSR-PP modules and DnResNet modules can be used to tackle multiple SR problems in the multiple tracks of NTIRE 2018 challenge.

### 3.3.4 Results of NTIRE 2018 Challenge

We have submitted enhanced images of DIV2K test data set to NTIRE 2018 challenge, Tracks 1, 2, and 3. Table 3.3 shows PSNR, SSIM and run time results for the top nine teams including our team using our proposed EMBSR method. Our team won the ninth place out of 24 teams with PSNR 25.331, SSIM 0.7026, and run time 2.52. Note that PSNR difference between the 1st place and ours was 0.124 dB and SSIM difference was 0.0062, but we achieved these results with the fastest run time among all top nine teams. Figure 3.5 shows qualitative results for bicubic interpolation, EDSR, and our EMBSR. Both EDSR and EMBSR yielded similarly good results, but EMBSR yielded higher PSNR than EDSR with slightly sharper images for some examples (see  $0820 \times 8$  from DIV2K results).

Table 3.3: Preliminary results of NTIRE 2018 challenge, Track 1,  $\times 8$  bicubic downsampling (PSNR in dB).

Method	PSNR	SSIM	Run Time
1st method	25.455	0.7088	50
2nd method	25.433	0.7067	20
3rd method	25.428	0.7055	6.75
4th method	25.415	0.7068	11.65
5th method	25.360	0.7031	7.31
6th method	25.356	0.7037	6.99
7th method	25.347	0.7023	5.03
8th method	25.338	0.7037	14.52
<b>Ours</b>	<b>25.331</b>	<b>0.7026</b>	2.52

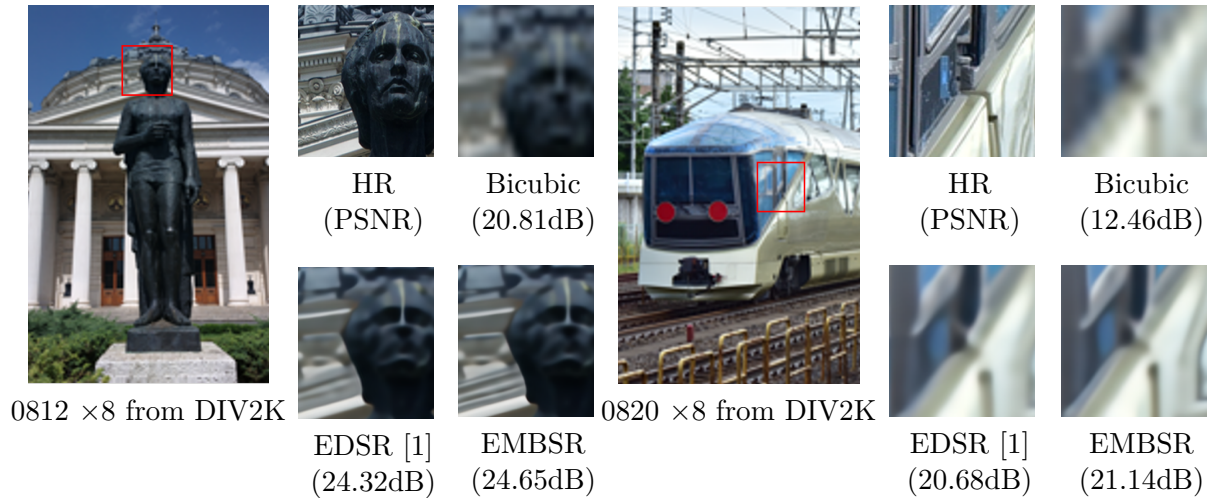


Figure 3.5: SR results of Track 1 in NTIRE 2018 challenge (bicubic downsampling  $\times 8$ ). Our EMBSR yielded better PSNR and slightly sharper images than EDSR.

Table 3.4: Preliminary results of NTIRE 2018 challenge, Track 2,  $\times 4$  unknown downsampling with mild blur and noise (PSNR in dB).

Method	PSNR	SSIM
1st method	24.238	0.6186
<b>Ours</b>	<b>24.106</b>	<b>0.6124</b>
3rd method	24.028	0.6108

Our proposed EMBSR methods achieved excellent performance in Tracks 2 and 3. Table 3.4 shows PSNR and SSIM results for the top three teams including our team for Track 2, unknown  $\times 4$  downsampling with image degradation due to mild blur and noise. Our team won the 2nd place out of 18 teams with PSNR 24.106 and SSIM 0.6124 in Track 2. Figure 3.6 shows qualitative results for bicubic interpolation, EDSR, and our EMBSR. Our EMBSR yielded significantly better image quality than EDSR quantitatively (Table 3.4) and qualitatively (Figure 3.6).

Table 3.5 shows PSNR and SSIM results for the top three teams including our team for Track 3, unknown  $\times 4$  downsampling with image degradation due to difficult blur and noise. Our team won the 3rd place out of 18 teams with PSNR 22.569 and SSIM 0.5420 in Track 3. Figure 3.7 shows qualitative results for bicubic interpolation, EDSR, and our EMBSR. EDSR does not seem to deal with multiple problems (SR, denoising, deblurring) well while our EMBSR efficiently tackled SR problem with multiple sources of image degradations. It seems that modular approach allows to use appropriate networks for different problems for improved performance.



Figure 3.6: SR results of Track 2 in NTIRE 2018 challenge (unknown downsampling  $\times 4$  with mild blur and noise). Our EMBSR yielded superior PSNR and image quality to EDSR and efficiently tackled SR problem with mild image degradation.



Figure 3.7: SR results of Track 3 in NTIRE 2018 challenge (unknown downsampling  $\times 4$  with mild blur and noise). Our EMBSR yielded superior PSNR and image quality to EDSR. EDSR does not seem to deal with multiple problems (SR, denoising, deblurring) well while our EMBSR efficiently tackled SR problem with multiple sources of image degradation.

Table 3.5: Preliminary results of NTIRE 2018 challenge, Track 3,  $\times 4$  unknown downsampling with difficult blur and noise (PSNR in dB).

<b>Method</b>	<b>PSNR</b>	<b>SSIM</b>
1st method	22.887	0.5580
2nd method	22.690	0.5458
<b>Ours</b>	<b>22.569</b>	<b>0.5420</b>

## CHAPTER IV

---

# Conclusion

---

In this study, We investigated Robot grasping detection and Image super-resolution using Deep learning. Robot grasping detection has shown that image resolution is better when resizing from 224 to 360. We also demonstrated that high accuracy of our proposed methods with our proposed learning-based, fully automatic vision-robot calibration method yielded 90% success rate in robotic grasping tasks with challenging small objects.

We proposed an efficient module based on single image super resolution network (EMBSR) using SR module (EDSR-PP) and denoising module (DnResNet). Modular approach allowed us to train our networks efficiently for multiple SR problems by recycling trained networks, to use modular ensemble for improved performance, and to deal with multiple sources of image degradation efficiently. We also proposed EDSR-PP, an improved version of previous ESDR by incorporating pyramid pooling so that global as well as local context information can be utilized. Lastly, we proposed a novel denoising / deblurring residual convolutional network (DnResNet) using our residual blocks based on DnCNN. The effectiveness of our proposed methods for multiple SR problems with mixed image degradation sources was demonstrated with NTIRE 2018 challenge by winning the 2nd place of Track 2, the 3rd place of Track 3, and the ninth place of Track 1 with the fastest run time.

As a results, we investigated the importance of resolution in robot grasping detection and how to efficiently use the network in Image Super-resolution. Through our pre-reaserch, we have contributed to Robot grasping detection using Super-Resolution.

---

## References

---

- [1] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee, “Enhanced Deep Residual Networks for Single Image Super-Resolution,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2017, pp. 1132–1140, IEEE. ix, x, 4, 5, 6, 22, 25, 27, 28, 29, 30, 31, 32
- [2] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang, “Learning Deep CNN Denoiser Prior for Image Restoration,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3929–3938. ix, 6, 22, 26, 28, 29
- [3] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 105–114. ix, 4, 21, 27
- [4] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang, “Learning a Deep Convolutional Network for Image Super-Resolution,” in *European Conference on Computer Vision (ECCV)*, 2014, pp. 184–199. x, 4, 21, 28, 29
- [5] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee, “Accurate image super-resolution using very deep convolutional networks,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1646–1654. x, 4, 21, 28, 29

---

**REFERENCES**

- [6] A Sahbani, S El-Khoury, and P Bidaud, “An overview of 3D object grasp synthesis algorithms,” *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326–336, Mar. 2012. 2
- [7] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic, “Data-Driven Grasp Synthesis—A Survey,” *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, Mar. 2014. 2, 8
- [8] Ashutosh Saxena, Justin Driemeyer, and Andrew Y Ng, “Robotic grasping of novel objects using vision,” *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, Feb. 2008. 2, 8
- [9] Jeannette Bohg and Danica Kragic, “Learning grasping points with shape context,” *Robotics and Autonomous Systems*, vol. 58, no. 4, pp. 362–377, Apr. 2010. 2
- [10] Yun Jiang, Stephen Moseson, and Ashutosh Saxena, “Efficient grasping from RGBD images: Learning using a new rectangle representation,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 3304–3311. 2, 8, 10, 18
- [11] A Krizhevsky, I Sutskever, and G E Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105. 2, 9, 10, 13
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep Residual Learning for Image Recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. 2, 9, 14
- [13] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Advances in Neural Information Processing Systems 28*, 2015, pp. 91–99. 2, 10
- [14] Joseph Redmon and Ali Farhadi, “YOLO9000: Better, Faster, Stronger,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517–6525. 2, 10, 11, 12, 14
- [15] Ian Lenz, Honglak Lee, and Ashutosh Saxena, “Deep Learning for Detecting Robotic Grasps,” in *Robotics: Science and Systems IX*, June 2013, p. P12. 2, 8, 16, 17, 18
- [16] Ian Lenz, Honglak Lee, and Ashutosh Saxena, “Deep learning for detecting robotic grasps,” *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, Apr. 2015. 2, 8, 10, 16, 17



---

REFERENCES

- [17] J Redmon and A Angelova, “Real-time grasp detection using convolutional neural networks,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1316–1322. 3, 9, 10, 11, 13, 16, 17, 18
- [18] Fu-Jen Chu, Ruinian Xu, and Patricio A Vela, “Real-World Multiobject, Multigrasp Detection,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3355–3362, Oct. 2018. 3, 4, 10, 13, 14, 18
- [19] Sung Cheol Park, Min Kyu Park, and Moon Gi Kang, “Super-resolution image reconstruction: a technical overview,” *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, May 2003. 4
- [20] D Robinson and P Milanfar, “Statistical performance analysis of super-resolution,” *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1413–1428, May 2006. 4
- [21] Se Young Chun and Jeffrey A Fessler, “Noise properties of motion-compensated tomographic image reconstruction methods,” *IEEE transactions on medical imaging*, vol. 32, no. 2, pp. 141–152, Feb. 2013. 4
- [22] William T Freeman, Egon C Pasztor, and Owen T Carmichael, “Learning low-level vision,” *International Journal of Computer Vision*, vol. 40, no. 1, pp. 25–47, Oct. 2000. 4
- [23] Daniel Glasner, Shai Bagon, and Michal Irani, “Super-resolution from a single image,” in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 349–356. 4
- [24] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. 4
- [25] Eirikur Agustsson and Radu Timofte, “NTIRE 2017 Challenge on Single Image Super-Resolution - Dataset and Study,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, 2017, pp. 1122–1131. 5, 6, 22, 28
- [26] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang, “Deep laplacian pyramid networks for fast and accurate super-resolution,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 624–632. 6, 21
- [27] Kaibing Zhang, Dacheng Tao, Xinbo Gao, Xuelong Li, and Zenggang Xiong, “Learning Multiple Linear Mappings for Efficient Single Image Super-Resolution,” *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 846–861, Jan. 2015. 6

---

**REFERENCES**

- [28] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia, “Pyramid scene parsing network,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2881–2890. 6, 25
- [29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. 6, 21
- [30] Zhichao Wang, Zhiqi Li, Bin Wang, and Hong Liu, “Robot grasp detection using multi-modal deep convolutional neural networks,” *Advances in Mechanical Engineering*, vol. 8, no. 9, pp. 1–12, Sept. 2016. 9
- [31] Jeffrey Mahler, Jacky Liang, Sherdil Niyaz, Michael Laskey, Richard Doan, Xinyu Liu, Juan Aparicio Ojea, and Ken Goldberg, “Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics,” in *Robotics: Science and Systems (RSS)*, 2017, pp. 1–10. 9
- [32] R Girshick, J Donahue, and T Darrell, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587. 9
- [33] Ross Girshick, “Fast R-CNN,” in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448. 9
- [34] Sulabh Kumra and Christopher Kanan, “Robotic grasp detection using deep convolutional neural networks,” in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 769–776. 9, 16, 18
- [35] Matthew D Zeiler and Rob Fergus, “Visualizing and understanding convolutional networks,” in *European Conference on Computer Vision (ECCV)*, 2014, pp. 818–833. 10
- [36] Di Guo, Fuchun Sun, Huaping Liu, Tao Kong, Bin Fang, and Ning Xi, “A hybrid deep architecture for robotic grasp detection,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1609–1614. 10, 12, 13, 14, 18
- [37] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788. 10, 11

REFERENCES

---

- [38] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, “SSD: Single Shot MultiBox Detector,” in *European Conference on Computer Vision (ECCV)*, 2016, pp. 21–37. 10
- [39] Umar Asif, Jianbin Tang, and Stefan Herrer, “GraspNet: An Efficient Convolutional Neural Network for Real-time Grasp Detection for Low-powered Devices,” in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018, pp. 4875–4882. 10
- [40] Karen Simonyan and Andrew Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014. 13
- [41] Robert Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996. 15
- [42] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007. 22
- [43] Viren Jain and Sebastian Seung, “Natural image denoising with convolutional networks,” in *Advances in Neural Information Processing Systems 21 (NIPS)*, 2009, pp. 769–776. 22
- [44] Harold C Burger, Christian J Schuler, and Stefan Harmeling, “Image denoising: Can plain neural networks compete with BM3D?,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 2392–2399. 22
- [45] Junyuan Xie, Linli Xu, and Enhong Chen, “Image denoising and inpainting with deep neural networks,” in *Advances in Neural Information Processing Systems 25 (NIPS)*, 2012, pp. 341–349. 22
- [46] Stamatios Lefkimmiatis, “Non-local Color Image Denoising with Convolutional Neural Networks,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5882–5891. 22