UTILIZING SUPER-RESOLUTION FOR ENHANCED AUTOMOTIVE RADAR OBJECT DETECTION

Asish kumar Mishra[⋆] *Kanishka Tyagi*[†] *Deepak Mishra*[⋆]

[⋆] Department of Avionics, Indian Institute of Space Science and Technology
Thiruvananthapuram, Kerala, India
[†]Aptiv Advance Research Center
Agoura Hills, California, USA

ABSTRACT

In recent years, automotive radar has become an integral part of the advanced safety sensor stack. Although radar gives a significant advantage over a camera or Lidar, it suffers from poor angular resolution, unwanted noises and significant object smearing across the angular bins, making radar-based object detection challenging. We propose a novel radar-based object detection utilizing a deep learning-based super-resolution (DLSR) model. Due to the unavailability of low-high resolution radar data pair, we first simulate the data to train a DLSR model. We develop a framework that feds a low-resolution radar dataset (called CRUW dataset) into the trained DLSR model pipeline to train a radar-based deep object detection classifier. The proposed framework achieves an 80% accuracy on object classification for the CRUW dataset and has a lower computational footprint, making it an ideal candidate for real-time implementation on edge devices used in autonomous driving applications.

Index Terms— automotive radar, radar object detection

1. INTRODUCTION

Automotive radar is a critical part of the advanced safety sensor stack. Automotive radar sensors provide accurate distance and velocity information and are highly robust to light and weather conditions (e.g., fog, rain, and snow). Radar spectra provide information in four dimensions: range, doppler, azimuth, and elevation. Most existing machine learning (ML) models for frequency-modulated continuous-wave radar (FMCW), especially automotive radars, do not rely on low-level time series data. Moreover, for front-looking FMCW radars, multi-target issues, background noises, and object smearing make it challenging to use them for perception tasks.

In this work, we use the idea of image super-resolution onto low-level radar images and develop a two-step strategy to perform object detection from the radar images. In the first step, we train a deep learning super-resolution model (DLSR) to generate high-resolution (HR) radar heat maps from low-resolution (LR). Accounting for the unavailability of proper datasets consisting of LR-HR radar heat map pairs, the DLSR model is trained on simulated radar data. We feed in the LR and HR heat map in the second step and perform object detection. The object detection block is not a single deep network but a unique approach involving a deep network, as will be explained further in the paper. Figure. 1 depicts the block-level diagram of the idea proposed in this paper.

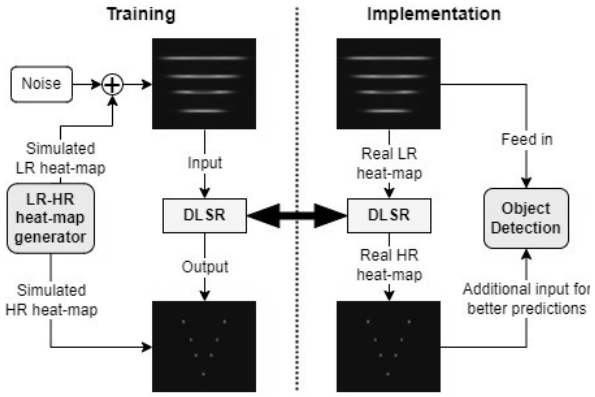

Fig. 1. Block level flowchart of the proposed work

Unlike cameras, where the object has a definite geometric boundary, low-level radar images are spectra with several peaks and minima. Due to this, image-based SR models cannot be directly utilized for radar-based SR methods. Though certain traditional approaches [6], [18],[17], [8] exist for radar SR, they are implemented on raw radar signals and time consuming to generate HR heat maps. The deep-learning-based radar SR approaches in [1], [5], [3] do not provide public dataset. Consequently, due to the unavailability of an adequate dataset, we simulated the radar dataset and released it. Many traditional radar detection level based approaches [2], [12] [4] for object detection exists. However, it is not real-time and uses detection level radar data [10]. Few studies have been carried out for radar-based object detection using

deep learning. [14] makes use of multiple raw radar signals to detect a target's presence. [16] propose a method for detecting and classifying objects in Radar heat maps through sensor fusion with a camera. [15] propose the idea of radar SR through the teacher-student model. To the best of our knowledge, none of the studies uses radar-only data for object detection.

There are **two main contributions** of our paper. First, we generated three types of dataset - Simulated LR-HR pair of RADAR maps, low-high pair of the CRUW dataset[15], and segmented image patches from the CRUW dataset - CRUW-Seg dataset. Second, we implemented a novel Radar object detection technique using Radar Super-resolution without using any other sensor or time-series RADAR data or Doppler domain.

2. PROPOSED ALGORITHM

Figure. 2 shows the novel approach proposed by this paper for implementing radar Object Detection. We obtained DLSR using a 3-layered encoder-decoder architecture, trained on LR-HR pair of simulated radar heat map. The trained model gave a corresponding HR radar heat map, eliminating most of the noises. The resultant HR radar images are thresholded (we choose 0.05) and used as a filter to separate the region of interest from the low-resolution images through masking. The mask images cover fixed azimuthal bins for various objects that can be extracted using a fixed rectangular mask. The masking operation helps eliminate the background noise from the LR images and focus on the object of interest. It is imperative to note that the HR image feature is critical for object classification that cannot be achieved on the HR radar image itself. Therefore, we use the HR image to eliminate the LR images' background noise.

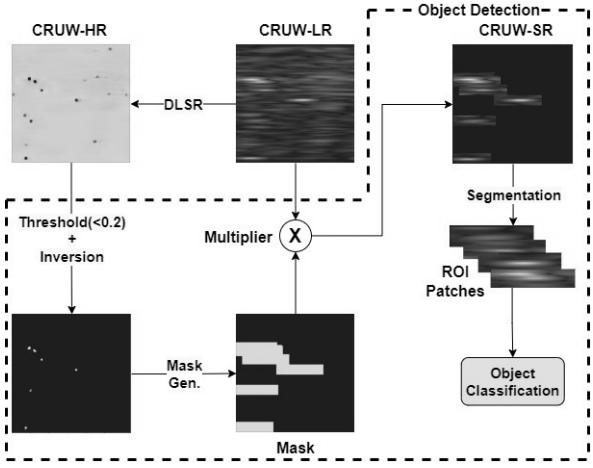

Fig. 2. Flowchart of the proposed framework. DLSR is trained to generate HR radar heat maps from LR heat maps. Object detection uses SR images to extract the region-of-interest patches from the radar heat maps.

Simulation Parameters	Values
Range span	$[0, 100]$ m
Azimuth span	$[-5°, 5°]$
Range Resolution	0.097 m
Azimuth Resolution (LR,HR)	$3.5°$, $0.0097°$
Original Pixel Resolution	512×512
Original Object Shape(HR)	20×20
Smearing Function Used	Sinc Square
Main Lobe Width of Smearing Function	$3.5°$ (179 pixels)
Total Number of Side lobes taken	4
Number of objects (Sparse,Dense)	4,10

Table 1. Values for the simulation parameters

The HR mask image is a binary image with 0s (absence of object) and 1's (presence of the object). When the LR radar image is multiplied (element-wise) with these mask images, we obtain an SR image focusing only on areas with a high probability of object detection. The SR heat maps can be passed to an object detection model; however, this is not a feasible approach due to extreme class imbalance. In order to solve this, we segment rectangular patches from the SR heat maps that are ready for object classification. We save the position of the patches present in SR heat maps for later use to localize the object.

In our work, we have four classes (car, cyclist, pedestrian and no-class) based on CRUW dataset. The object classifier is a 13-layer deep network that is trained using Adam optimizer. The objects, after being classified, are identified in the LR radar image through saved spatial parameters from the SR heat maps.

3. EXPERIMENTAL SETUP AND RESULTS

3.1. Simulation details

In order to generate the HR radar map, a fixed range and angle span are assumed as per the [8], and small random patches (mainly 20×20 pixels patches), corresponding to the objects in the scene, are introduced in the heat map. Variation in the shape and size of the object patches is also introduced up to $\pm 200\%$ of the area of the original object patches to consider object variety. Similarly, the simulated LR radar image is an HR heat map convolved with a pre-defined sinc square function (equivalent to a reflection in a real-life environment) along with added noises. Ideally, the width of the main lobe of the sinc square function can never exceed the angular beam width(assumed as $3.5°$ referring to SFSBA [8]) of the antenna. Therefore, while carrying out the simulation, the main lobe width of the sinc square function is taken as $3.5°$ as opposed to the total width of the image as $10°$ ($[-5°, 5°]$). Table. 1 lists out the simulation parameters and their values during the entire simulation process.

We simulate a highway and an urban scenario as sparse and dense environments. A total of 3000 gray-scale low-high

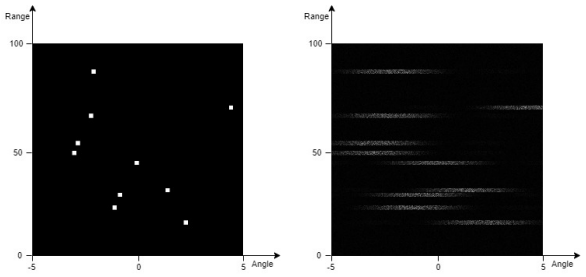

Fig. 3. Polar Radar images generated from the simulation. Left - GT/HR Radar Image, Right - LR Radar Image with added Noise

image pairs are generated for each of the two scenes with a pixel dimension of 512×512 and down-sampled to 128×128 pixels for training so that the parameters of the network reducFigure. 3 shows an example of simulated HR-LR pair. Our study uses Gaussian and speckle noise since they capture a much more realistic background noise and give better results than other available noise distributions.

3.2. CRUW LR-HR pair and CRUW-Seg data generation

To generate the LR-HR pair for the CRUW dataset, we feed 36800 LR radar heat maps of size 128×128 into a trained DLSR network to generate HR radar heat maps. Figure. 2 shows an example of a generated LR-HR pair of CRUW data. Due to the lack of available training data for object detection on masked HR images, segmented image patches are used to train the object classifier. The object location in the image generates the objects for the three classes in the CRUW dataset: car, cyclist, and pedestrian. Since the smearing width does not vary significantly for these three classes, rectangular patches of the same dimension (50×10 pixels) are chosen for all the classes. To generate the examples for the background(none) class, four random samples are taken from each image around which rectangular patches are segmented. The samples are chosen so that the Intersection-over-Union for the rectangular patches is at most 0.33 to ensure uniqueness and remove poor examples. The CRUW-Seg data contains 186 thousand segmented 50×10 patches divided across four classes.

3.3. Data Augmentation

At the DLSR stage of the framework, a problematic issue of an all-zero output problem arises. As most of the input and output images are zero or near-zero values, it is challenging to train the DLSR model. Most model parameters are exposed to zero input values during training, leading to no learning. Even though metrics and loss functions specific to class imbalance are used, more is needed for the model to come out of the all-zero output issue. Consequently, the model predicts a poor HR heat map.

We perform a pixel inversion technique to force the model to take non-zero values to solve the issue. To avoid direct input-output mapping on the same values, it is necessary to invert the pixel values for either input or output maps. In doing this, the model weights take non-zero values to converge to a better prediction model.

3.4. DLSR stage

A simple three-layer encoder-decoder network architecture is followed for implementing the Super-resolution (DLSR stage). A block of the architecture encoder contained two convolution layers of 3×3 kernel with a relu activation followed by a MaxPooling layer. The number of filters remained the same for both convolution layers inside the block. The encoder has three blocks consisting of 64, 128, and 256 filters, respectively, with the last block omitting the MaxPooling Layer. Similarly, a block in the decoder consisted of an Up-Sampling Layer followed by two convolution layers of 3×3 filter with a relu activation. The decoder contained two blocks with 64 and 128 filters, respectively, followed by a softmax layer to get the desired output image. This architecture gives the best performance after conducting experiments with slight variations in the network design. MSE loss function is used in the DLSR stage along with L_1 regularization and Adagrad optimizer with a learning rate of 0.01. Finally, we choose the F1 score as an evaluation metric.

Figure. 4 shows the results of the DLSR stage on two randomly selected samples from CRUW[15] dataset. It shows that the model performs Super-resolution well by eliminating unnecessary noises and reducing the smear. It can also be seen that points in the Radar map corresponding to the objects in the camera appear as slightly darker spots in the high-resolution heat map, thus proposing the presence of an object which ultimately helps us carry out object detection.

Because of the unavailability of a public dataset containing Radar LR-HR pair, it is directly not possible to get a test accuracy score. Therefore, we devised a fundamental approach by which we can understand the effectiveness of the model. Since the CRUW dataset gives the object's location, we take a patch of the neighborhood around this location and find the maximum pixel value in this patch. The maximum value is then compared with a pixel threshold hyper-parameter for scoring the presence of an object. If the maximum value exceeds the pixel threshold, we score it as 1; otherwise, we score it as 0, and the overall mean value is taken. This mean value is found for both the CRUW dataset and the Super-resolution image, and a ratio is then taken as the accuracy score. On choosing a threshold value of 0.5 and a neighborhood of 11×11 around the object's location, we get an accuracy score of 95.36%.

3.5. Object Classification Stage

The deep convnet used in the object classification stage consists of five repeated stacks of convolution layer, batch-

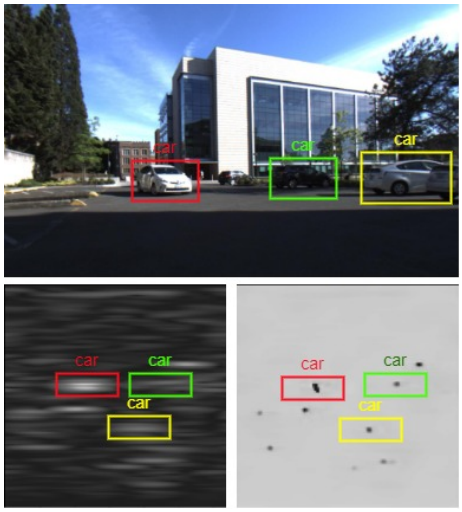

Fig. 4. Top - Camera image of the corresponding scene, Left bottom - Original CRUW Low-resolution Image. Right bottom - Final CRUW high-resolution images as predicted by the Super-resolution model.

normalization, ReLU activation, and MaxPooling layer, followed by two linear layers with ReLU activation and a sigmoid output layer. All the convolution layers have a 3×3 kernel and a 'same' zero padding, whereas all the MaxPooling layers have 2×2 kernel and a stride of 2 with no zero-padding. The filters in the five convolution layers starting from the input are 2, 4, 8, 16, and 32, respectively. The number of neurons in the linear layers is 40 and 16, respectively, followed by a 4-neuron output layer. Cross-entropy is used as a loss function and AdamW optimizer with a linearly varying learning rate from 0.01 to 0.008. The accuracy metric is used as the evaluation criterion.

We implement two variations of object detection: 3-class and 4-class (including background class). The 3-class classifier though not accurate, is justified as the DLSR stage eliminates most of the noises. Table. 2 compares the accuracy and number of parameters with other state-of-the-art object classifiers. The other classifiers in Table. 2 are 2-class classifiers as opposed to ours. In our case, the DLSR stage, similar to a 2-class classifier(differentiating objects from noises), achieves a 95% accuracy. Therefore, our model performs better than other radar-based object detection models.

Compared to [15] and [16], our proposed model achieves a lesser accuracy in object detection. This is not a fair comparison since other studies have used extra sensors like lidar and camera or continuous time-series outputs in the range-doppler domain. Our model performs better than other vision-based object detectors like SSDs, which give accuracy below 10% on the CRUW dataset as per our implementation. We suspect that the over-fitting of the model due to the lesser number of training examples may be accountable for such lesser accuracy. For vision-based object detectors on radar data, we observed that the detectors give three times more accuracy when trained and tested on the masked SR images than directly being fed low-resolution CRUW images. This experiment further bolsters the effectiveness of our approach. The total number of parameters for both models amounts to 1.11M parameters. Additionally, the total size of both the models combined is approximately 13MB, and the prediction takes less than ten milliseconds on a 32-GB GPU, making the approach very effective for real-time hardware implementation.

Network	Val Acc(%)	Parameter Count
Ours(3-class)	80.0	14.7k
Ours(4-class)	75.0	14.8k
RaDICaL(linear)* [9]	83.1	1.7M
RaDICaL(log)* [9]	80.0	1.7M
MobileNetv2* [11]	85.1	2.23M
ResNet50* [7]	84.08	23.5M
VGG16* [13]	50.0	33.6M

*Implemented as a 2-class classifier [9]

Table 2. Comparision of accuracy and number of parameters for the object classification stage

3.6. Miscellaneous

The data is split into 70% training, 20% validation, and 10% testing, along with five-fold validation. We use the PyTorch framework on a 32GB NVIDIA Tesla V100 GPU. The DLSR network is trained and validated with the simulated data in a train-to-validation ratio of 11:4 and tested on the complete CRUW dataset. For the object classification network, a train-to-test ratio of 4:1 is used out of the total CRUW-Seg data.

4. CONCLUSION

The work proposes a novel framework using super-resolution for detecting objects in radar images. We train a super-resolution model on simulated data to compensate for the scarcity of low-high resolution pairs of radar heat maps. The trained model generates the low-high resolution pair from the CRUW dataset, which is later used to generate ROI patches for object classification. Our work is unique because we only use radar sensors and a combination of simulated and real data in our pipeline. Additionally, the low computational overhead of the whole framework makes it an ideal candidate for real-life applications where low computation and low-memory usage are required. The present study is limited because it is a two-stage framework rather than an end-to-end trainable network that can take a low-resolution radar image, perform super-resolution, and then object classification. Since the radar data is sparse, the dataset for training the deep learning model is highly imbalanced, and therefore better techniques have to be employed to avoid over-fitting.

5. REFERENCES

[1] Karim Armanious, Sherif Abdulatif, Fady Aziz, Urs Schneider, and Bin Yang. An adversarial super-resolution remedy for radar design trade-offs. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5. IEEE, 2019.

[2] Jack Capon. High-resolution frequency-wavenumber spectrum analysis. *Proceedings of the IEEE*, 57(8):1408–1418, 1969.

[3] Yu-Zhang Chen, Tsung-Jung Liu, and Kuan-Hsien Liu. Super-resolution of satellite images by two-dimensional rrdb and edge-enhancement generative adversarial network. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1825–1829. IEEE, 2022.

[4] Seunghoon Cho, Heemang Song, Kyung-Jin You, and Hyun-Chool Shin. A new direction-of-arrival estimation method using automotive radar sensor arrays. *International Journal of Distributed Sensor Networks*, 13(6):1550147717713628, 2017.

[5] Andrew Geiss and Joseph C Hardin. Radar super resolution using a deep convolutional neural network. *Journal of Atmospheric and Oceanic Technology*, 37(12):2197–2207, 2020.

[6] Gene H Golub, Per Christian Hansen, and Dianne P O'Leary. Tikhonov regularization and total least squares. *SIAM journal on matrix analysis and applications*, 21(1):185–194, 1999.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Weibo Huo, Qiping Zhang, Yin Zhang, Yongchao Zhang, Yulin Huang, and Jianyu Yang. A superfast super-resolution method for radar forward-looking imaging. *Sensors*, 21(3):817, 2021.

[9] Teck-Yian Lim, Spencer A Markowitz, and Minh N Do. Radical: A synchronized fmcw radar, depth, imu and rgb camera data dataset with low-level fmcw radar signals. *IEEE Journal of Selected Topics in Signal Processing*, 15(4):941–953, 2021.

[10] Mark A Richards. *Fundamentals of radar signal processing*. McGraw-Hill Education, 2014.

[11] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[12] Ralph Schmidt. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 34(3):276–280, 1986.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[14] Li Wang, Jun Tang, and Qingmin Liao. A study on radar target detection based on deep neural networks. *IEEE Sensors Letters*, 3(3):1–4, 2019.

[15] Yizhou Wang, Zhongyu Jiang, Xiangyu Gao, Jenq-Neng Hwang, Guanbin Xing, and Hui Liu. Rodnet: Radar object detection using cross-modal supervision. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 504–513, January 2021.

[16] Yizhou Wang, Gaoang Wang, Hung-Min Hsu, Hui Liu, and Jenq-Neng Hwang. Rethinking of radar's role: A camera-radar dataset and systematic annotator via coordinate alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2815–2824, 2021.

[17] Yin Zhang, Qiping Zhang, Yongchao Zhang, Jifang Pei, Yulin Huang, and Jianyu Yang. Fast split bregman based deconvolution algorithm for airborne radar imaging. *Remote Sensing*, 12(11):1747, 2020.

[18] Yongchao Zhang, Yin Zhang, Yulin Huang, Wenchao Li, and Jianyu Yang. Angular superresolution for scanning radar with improved regularized iterative adaptive approach. *IEEE Geoscience and Remote Sensing Letters*, 13(6):846–850, 2016.