Improving Millimeter Wave Radar Perception with Deep Learning

Junfeng Guan Sohrab Madani
ECE 544 Project Report
jguan8@illinois.edu smadani2@illinois.edu

1. Introduction and Project Descriptions

Since the past few years, AI-Powered autonomy revolution in the automotive industry has attracted great attention worldwide. It is believed that in the not-too-distant future, fully autonomous vehicles will be the norm rather than the exception, redefining mobility in our daily lives. With deep learning widely applied on sensor data, self-driving cars are able to localize and map objects, understand the environment, and eventually make right decisions. As the most essential task, previous works have demonstrated accurate object detection and classification based on sensor data, but they are limited to LiDARs and cameras. These optical sensors can produce high-resolution images, but they naturally fail in low visibility conditions such as fog, rain, and snow, because light beams are narrower than water droplets and snowflakes [10]. Such fundamental limitation of optical sensors is one of the major roadblocks to achieving the 5th SAE level of full automation [2]. On the contrary, Radar wave can propagate through small particles, which make it an possible alternate imaging solution in such inclement weather. Besides, radar also possess many other advantageous features. For example, it can directly measure the velocity of vehicles based on the Doppler shift of the reflected signal instead of going through cluster tracking among frames like LiDAR. With additional real-time speed information of objects, AI can potentially have a better perception than human drivers.

Although the low resolution of traditional automotive radar overshadows its advantages, the advent of mmWave (Millimeter-wave) technology makes it possible to have a reliable 3D imaging system in inclement weather with relatively higher resolution. Along with good propagation characteristics, mmWave also provides much wider bandwidth and enables miniature antenna arrays, which improves the distance and angle estimation respectively. Previous works has demonstrated sub-centimeter level imaging resolution for short range objects [11], but the fundamental difference in wavelength between radar (5 mm) and LiDAR (905 nm) cannot be easily overcome. Moreover, radar images are not as readable to non-experts, so in order to convince the automotive industry and general public that mmWave is a reliable imaging solution for autonomous driving, we should try to achieve LiDAR grade images, not only of comparable resolution, but also more perceptually intuitive to people.

In this project, we propose to develop high-quality and reliable imaging techniques for self-driving cars with mmWave radar. Specifically training neural networks to enhance the low-resolution and unreadable radar images to be similar to the LiDAR point cloud, which has been extensively and successfully used for self-driving perception. Ultimately achieving various crucial vision applications for autonomous vehicles like lane detection, object localization and identification with enhanced radar images only. To do this, we have to overcome challenges analyzed in the following section 2.

2. Challenges

2.1. Radar Imaging Primer

Radar generates images of objects by localizing point reflector models corresponds to the object. These point reflector models in the 3D space lie in voxels in a spherical coordinate determined by distance, azimuth and elevation angles. Distance is measured as the round-trip ToF (Time-of-Flight) of reflected radar signal times the speed of light. The azimuth and elevation angles can be either regarded as the beam steering angle of phased array antennas, or be estimated with DoA (Direction-of-Arrival) estimation algorithms such as beam-forming or MUSIC (Multiple Signal Classification). Notice that at further distance, the voxel size increases, so that long-range objects spread across much fewer voxels so that there shapes become more blurry. Besides, unlike the extremely narrow light beams, the cone-shape radar beams with sidelobes cause interference and leakage among voxels, which furthermore smears radar images and introduces noise. Last but not least, radar reflection tends to be more specular than the mostly scattering at optical frequencies. To sum up, ”edge detection” in radar images is very different from that of computer vision. Firstly, it needs to predict the high frequency information with only the low frequency data. Secondly, it needs to fill in the missing parts of objects.

2.2. Related Work

Previous works have attempted to adapt the optical camera-oriented Convolutional Neural Network (CNN) to its microwave counterpart to classify a single object in high-resolution 2D SAR (Synthetic Aperture Radar) image [4, 15, 6]. There has also been successful application of CNNs on recreating short range human body skeletons in 2D planes and 3D spaces from radar images by tracking 14 key points [21, 22]. However, the scope of these applications of neural networks are very restricted features and single object classification. Besides, their radar image are either of super high resolution generated by airborne geographical sensing synthetic aperture radars or within a short range where voxel resolution does not degrade too much. On the contrary, in our application of autonomous cars, we do not have as high resolution with way smaller antenna array at longer distance, but we need precisely traced boundaries with the information of size, shape, and orientation contained. Therefore, if we simply extend the layer and number of output neurons for fine grained feature detection to form boundaries, we might need a very deep neural network. Given that CNN structures haven't be well investigated and optimized for radar signal and images, starting from scratch could be extremely difficult. However, the emergence of GAN (Generative Adversarial Networks) and conditional GAN provides us great opportunity to transfer radar images from the domain of low resolution to the domain of higher resolution with sharp boundary and missing parts and details filled up.

2.3. Dataset Availability

Once we chose conditional GAN as the network structure, we came across the problem of dataset availability. Ideally, we should provide a large number of radar images of various environments that self-driving cars would experience, as well as the precisely traced object boundaries and orientations as the ground truth. However, we were not able to find any public dataset of 3D or even 2D radar images, especially street views for the self-driving car application. Also in contrast to the standard images generated by cameras and LiDARs from different manufactures, radar images generated by different system and algorithms have non-negligible variation. Thus we were initially forced to build our own dataset of 3D radar images with our custom-built 60 GHz mmWave radar imaging system. However, since our system was design to compensate the frame rate for lower cost and higher resolution, even collecting 500 frames continuously will take at least 41.67 hours. Not to mention that we also have to wisely choose plenty different locations to mimic various scenes. The time cost of generating this dataset is impractical. Besides, for the ground truth dataset we even have to perfectly sync our reference camera or LiDAR for ground truth with the radar, so that the radar images would share the same point of view as the photos and LiDAR point clouds. All of these overheads add up to a huge roadblock at the starting line of our project. Therefore, instead of building dataset from real radar images, we started to leverage the well-developed EM (Electromagnetic) simulation tools and the exquisitely measured waveform of our radar waveform to simulate radar reflections and feed the simulated radar signal to the same image processing algorithm we used to synthesize radar images. In order to make our synthesized scenes and images as similar to actual road environments as possible, we try to recover them from video recording of street view in datasets like the Cityscapes [3].

3. Method

3.1. Conditional GAN

Generative Adverserial Networks (or GANs) [7] have been widely used to generate images. The input to these networks can be text [13, 19], where images are generated according to some text or labels; or images [9, 16, 20, 18], where the network is trying to fill in some missing part of the input image, or translate the image to another domain. In our case, we are looking to generate images with accurate boundaries using low resolution images with missing parts as input. Conditional GANs have already proven successful in similar settings, such as in [12, 20], where the authors have used thermal images under low light conditions where there some parts of the image are missing to retrieve the human face boundaries and estimate its orientation. Another motivation behind using conditional GANs is that loss functions such as the L_2 and L_1 (i.e. loss functions that are equal the Euclidean or L_1 distance between the input and the output) which are the de facto standard loss functions for restoring images, render blurry images as output, which are not suitable for our application as they do not emphasize on boundaries. On the other hand, using conditional GAN, we were be able to motivate the loss function in GAN to learn to focus on the boundaries, by designing the ground to contain information mostly about the boundary of objects, as discussed in more detail the dataset section.

We have adopted the CGAN architecture from [9] to train and test our model. Denoting the input, ground truth, and noise by x, y, and z respectively, we can write the objective for a conditional GAN where the Generator and discriminator are G and D as

$$\begin{aligned} &\min_G \max_D \mathcal{V}_{CGAN}(D,G) = \mathbb{E}_x(\log(D(x|y))) \\ &\quad + \mathbb{E}_z(\log(1 - D(x, G(x|z)))). \end{aligned} \tag{1}$$

Here, $D()$ is the score that the discriminator gives for a certain input, and $G(x, z)$ is what the generator tries to generate given the input x and noise vector z. The point of having the noise vector, of course, it to avoid the problem

of over-fitting. The difference between the standard GAN and the conditional version is that in the latter everything is conditioned to y. This y could be anything we want, that adds extra information about what the output G should look like. In our setting, we call y the ground truth, as it contains information of real boundaries of the objects. Given y, D tries to maximize its output value when the input is real, and at the same time minimize it when the input is artificially generated by the G, the generator. The generator tries to generate an image that gets a high score from D, motivating it to generate images that are similar to real ones.

It is suggested in [pix2pix] that we can combine generator's task of trying to get a high score from discriminator with a pre-determined loss function, such as L_1. That is, one could change the objective to be

$$\begin{aligned} &\min_G \max_D \mathcal{V}_{CGAN}(D,G) = \\ &\mathbb{E}_x(\log(D(x|y))) \quad + \quad \mathbb{E}_z(\log(1 - D(x, G(x|z)))) \\ &+ \lambda \mathcal{L}_{L_1}(G(x,z)). \end{aligned} \tag{2}$$

The motivation behind this is to capture the low-frequency information using the L_1 loss, and motivate GAN to model high-frequency information, which in our case, will translate to more precision in identifying boundaries.

For the generator, was have adopted the architecture introduced in [14], which is an auto-encoder. Similar to most such architectures, U-net consists of a contracting path and an expansive path. The first layer contracting path is made of two successive 3×3 convolutions followed by a ReLu (rectified linear unit) and a 2×2 max-pooling layer without overlapping, which shrinks the size of its input by a factor of four. Let us call such a layer a down-sampling layer. This layer is then repeated multiple time, each time doubling the feature channels. The expansive path reverses this process: at each layer, the data is first up-sampled by a factor of two, the filtered with using a 2×2 window (2 by 2 convolution) halving the number of feature channels. At this point, a cropped version feature maps from the corresponding layer in the contracting path is forwarded to this layer (i.e. for layer k, the feature maps from layer $n - k$ are forwarded, assuming a total number of n layers) and are concatenated with the current feature maps. There are two points to be mentioned here; first, the cropping should happen because of the loss of edges that has happened due to successive convolutions. Second, and more importantly, this idea is used as a way to detour the bottleneck layer (the middle layer through which all the information should pass) by creating these skip connections between layers. One reason behind this choice of architecture for our problem is that this forwarding encourages the similar structure between the input and the output. Authors in [14] evaluated this architecture in a biomedical image segmentation problem its idea has, and in [9], it was adopted to generate a general translator of high-resolution output images to high resolution input images, which is quite similar to our setting, except that in our problem the input images are not high-resolution.

As for the discriminator, what we need is for the network to be able to identify local structures, in order to capture the properties of local objects (e.g. cars). Since the network is relying on the L_1 loss to guarantee the correctness of low-frequency information, it is possible to restrict the discriminator to only penalize structures that occur within a patch window of the image. In other words, the discriminator slides over the image looking at patches of size $n \times n$, and scoring each of them, and finally averaging over all patches to derive the final score. This structure has been dubbed patchGAN [9]. For our implementation, we chose n to be 70 where the image size was 256×256.

3.2. Dataset Generation

As analyzed in section 2, when exploiting the application of deep learning in the new field of radar images, the problem of dataset availability is inevitable. Between the two options of building our own dataset: collecting real radar images through experiments and processing synthesized radar reflection, we have to make a trade-off. Although synthesizing dataset might be the only feasible way to create a big enough training dataset, the compatibility of trained model to real radar images significantly depends on the closeness between synthesized and real radar reflection. Luckily, the transmission, propagation, and reflection of radar waveforms have been thoroughly studied so that they can be precisely modeled and simulated. Furthermore, with the help of advanced electromagnetic simulation softwares such as FEKO [1], we can even model the attenuation and specularity of various objects and surfaces. Therefore, a well-designed radar reflection synthesizing algorithm should be indistinguishable from read radar signal, so it is hopeful that the performance of a conditional GAN model trained with synthesized radar images should not degrade too much when we use real radar image for testing instead. We are also planning to mixing a smaller number of real radar images with the synthesized dataset for training and find the improvement of cGAN performance with respect to the portion of real and synthesized data.

The radar image synthesizing process can be further separated into four steps: scene generation, radar reflector modeling, radar signal simulation, and image processing. The last two steps are standard and straightforward, while the challenge lies in creating the 3D space distribution of point reflectors based on a realistic scene of automotives. We found two types of dataset that can be of great use for us. The first type of dataset is 2D video recording of street scenes from the view of a car, such as the Cityscapes dataset [3], while the second type of dataset is 3D CAD models of typical objects on the road. Good examples are the Deformable 3D Cuboid Model dataset and the 3D ShapeNets

dataset [5, 17]. These two types of datasets contains unique information and make great complement to the other.

Video recordings from the Cityscapes includes a wide range of practical urban road environment from the point of view of a car, but the 3D shape of objects cannot be recovered at all. Besides, we could only very roughly estimate the location of most common objects like vehicles and human, so that from the 2D pictures of street view, we can at most place 2D shapes at roughly estimated angle and distance. On the other hand, the 3D CAD models provide us with precise 3D distribution of point reflectors that consists typical objects on the road such as cars, trucks, cyclists, and pedestrians. We can also find out the effective reflectors in the sight from an arbitrary angle, and even the specularity of reflection off surfaces. However, it is not trivial to place models and compose a verisimilar 3D scene. In our experiments, we first tried to create scenes with 2D and 3D datasets separately. For the 2D dataset, we use vision-based object detection neural networks Mask R-CNN [8] to detect masks of objects and label them, and then roughly estimate and place 2D shapes of objects based on their size and pixel-wise position to recover scenes in 3D space. After that, we assign a number of point reflectors to every objects according to their shape and labels, and from there radar images can be synthesized. In this way we can potentially build a dataset with the same size of the Cityscapes, although the EM simulation takes time and the synthesized 3D radar images are of 2D shapes. However, we want to argue that if we project the 3D synthesized image back to onto a screen, the effect of distance estimation of objects on the output image can be ignored. We are also planning to combine the 2D and 3D datasets to compensate the shortcomings of using 2D video recordings alone, and synthesize a realistic 3D street scene dataset. We will start with finding the 3D model of the object type classified by the Mask R-CNN, together with a point of view, and even distance that best fits the object mask in the video recording. According to this more precise location and orientation estimation, we can place the 3D models at almost the same place as in the video. Followed by the effective point reflector assignment with the specularity into consideration, and radar signal and imaging simulation. We can synthesize very authentic 3D radar images only at the cost of computation time.

4. Experiments and Results

We first conducted experiments with our custom-built 60 GHz mmWave imaging system to collect real 3D radar images. Our system and the experiment setup is shown in figure 1. Figure 2 is an example of the radar image we generated along with the scene it corresponds to. Although we can infer the overall shape of the Jeep from the radar image, the specularity effect breaks the car body into sparse clusters of reflectors.

Figure 1: Custom-built 60 GHz mmWave radar imaging system

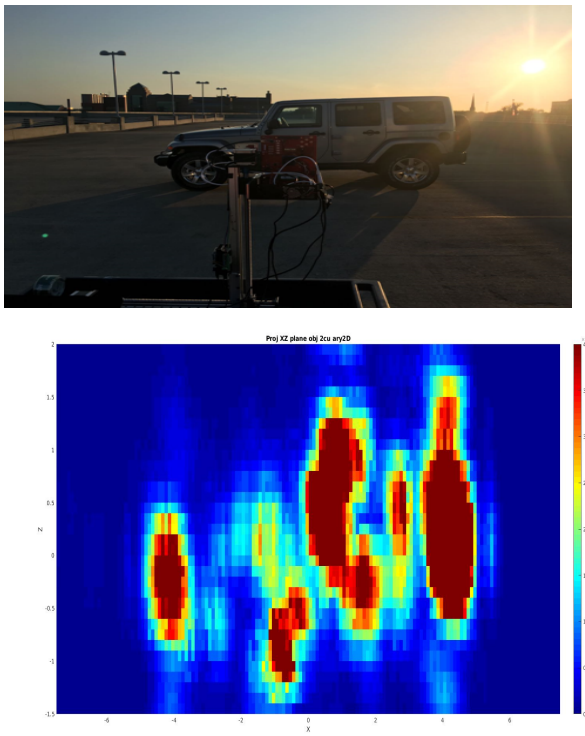


Figure 2: Real radar image of a 2015 Jeep Wrangler

Figure 3 and 4 demonstrate our 2D and 3D radar image synthesizing procedures. For 2D images generation, we start with video frames from the Cityscapes dataset [3], and

run the trained Mask R-CNN model to detect and label objects [8]. Mask R-CNN will also generate object masks which we will use to roughly recover the 3D scene and reflector model with. After that, we simulate radar signal and produce synthesized 2D radar images. At the same time, we extract high quality images of objects from the camera pictures as the ground truth for our conditional GAN model. For 3D image generation, we begin with 3D CAD models [5]. We first transform the mesh format model into point reflectors by filling the mesh surfaces. We then choose a reasonable point of view and transfer point reflector model into a spherical coordinate with the origin at our observing point. By selecting reflectors that are not blocked by others in front of them, we get the effective reflector distribution, and randomly choose a number of clusters to emulate the specularity effect. At the end, we implement the same radar simulation algorithms for the 2D case and generate realistic 3D radar images.

For our experiments, we ran our data one NVIDIA Tesla K80 GPU available on Google Cloud using CUDA 9.0 and PyTorch 0.4, along with a local Pascal GTX 1080 GPU using CUDA 10.0 and PyTorch 1.0. We used 1056 images of size 256×256 as for training, and 452 images for testing. The training phase takes about 11 hours to finish if the complete training dataset is used.

The first experiment we ran on the GPU was a toy example using blurred images as input. We took the images of cars, added some Gaussian noise and convolved the image with a two-dimensional $sinc$ function. We fed these images as input, and the original images to the network as ground truth. the results showed that the network is able to restore the boundaries of the cars with a high accuracy. The problem with this implementation was that the image was not gray-scaled, and we did not go through a precise simulation according the to underlying physics of mmWave radars to generate the images. The purpose of this step was to evaluate the feasibility of our idea of restoring the boundaries using GANs on a high level. The results from this experiments are shown in figure 5. It can be seen that the boundary of the car has been fairly accurately reconstructed. It is also noteworthy that the GAN was able to restore the side mirror of the car successfully, although there is no side mirror visible point in GAN.

In the second experiment, as mentioned in the dataset section, we synthesized a more realistic dataset that is very similar to radar images. In most cases, the GAN was able to restore the boundaries and the orientation of the cars 6. It has also learned to identify the cars some of whose parts are blocked, either because they appear near the edge of image, or appear behind other cars.

As mentioned before, one significant problem with radar images is that some objects in appear to be much weaker or stronger in power than others, depending on their angle with

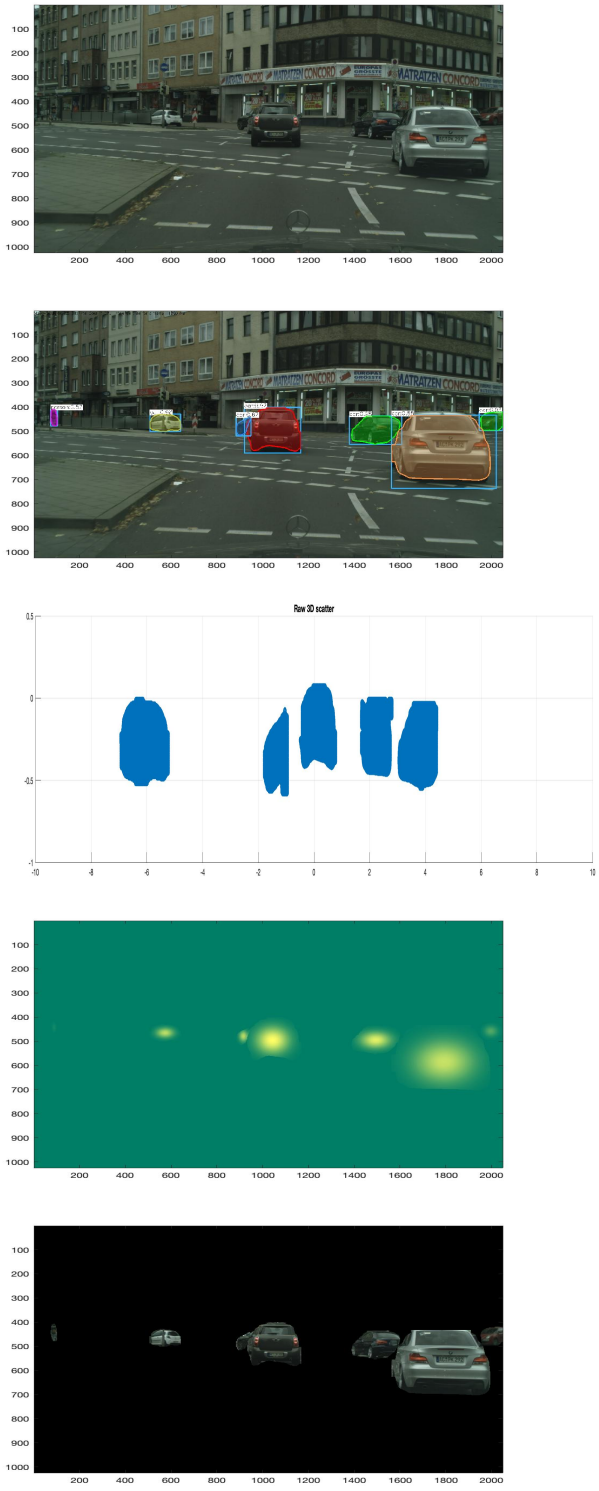


Figure 3: 2D radar image synthesizing procedure and result

the receiver mmWave antenna. The results from this experiment show that the GAN is able to learn this phenomenon

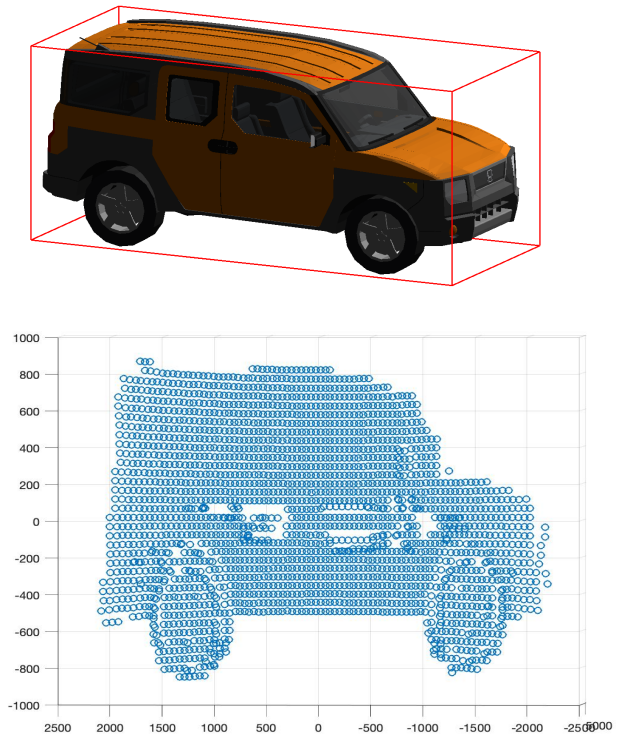


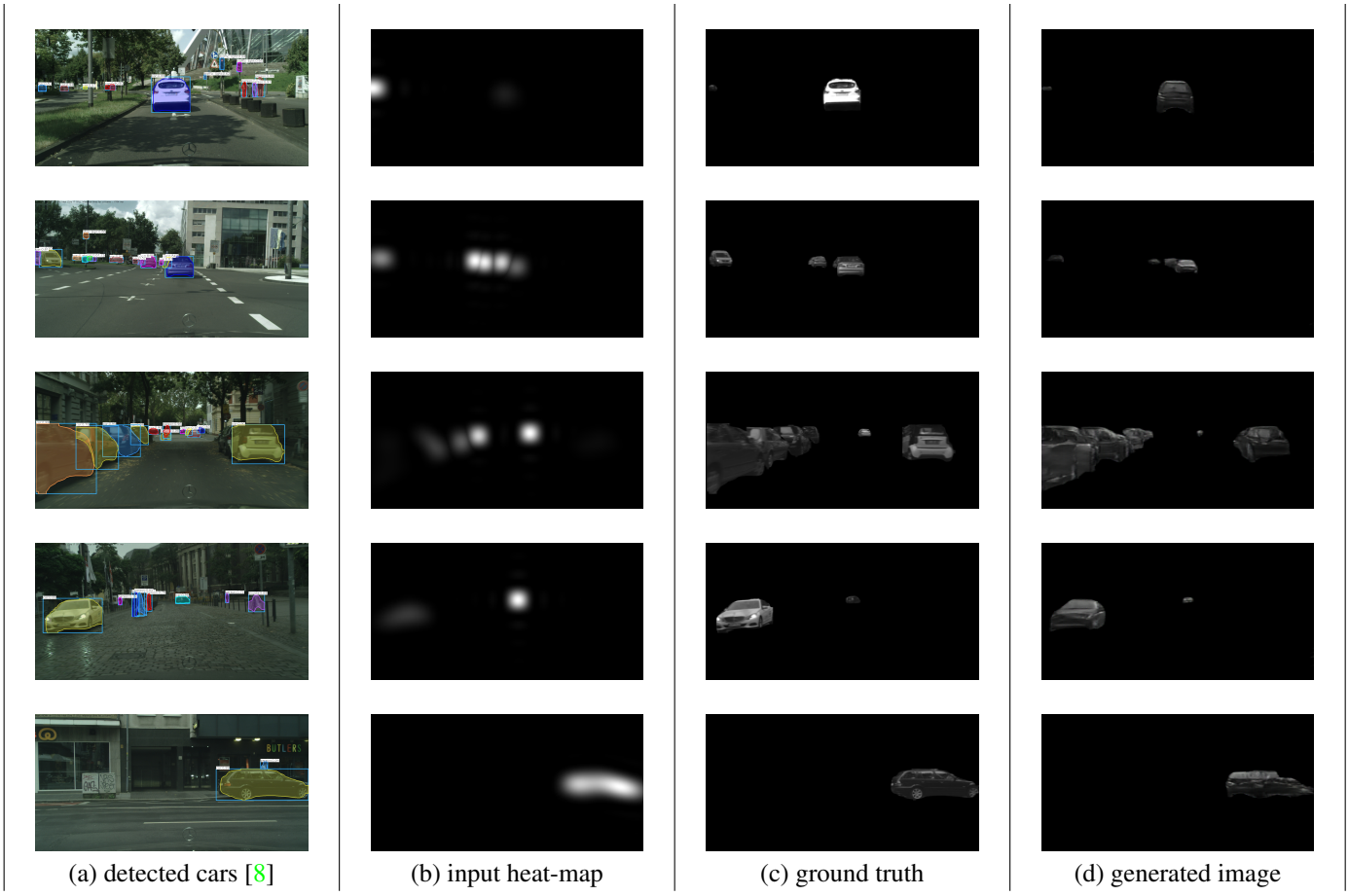
Figure 4: Custom-built 60 GHz mmWave radar imaging system

Figure 5: Result from the first experiment. The GAN has succesfully estimated the boundary of the car.

and compensate for it. As a result, it is able to identify the cars, even if the reflection from them is weak. For instance, in the fourth row of figure 6, the GAN successfully

5. Discussion and Conclusion

In this project we demonstrate an adapt of conditional GAN to radar images, aiming to improve the tradition low resolution, incomplete, and incomprehensible radar images in a way that they clearly depict the boundary and orientation of objects. GAN makes a great candidate for this specific application because it can successfully estimate high resolution information only low resolution data. However, as not been widely studied with neural networks, there is almost no public radar image dataset available, especially for the application of autonomous cars. Therefore, we also designed a data transformation and augmentation approach that leverages well-established vision and CAD model databases to synthesize realistic radar images. Experiments of 2D radar images have successfully generated enhanced radar images with precise boundaries and orientation with an 180 degree ambiguity, which can be easily fixed with the sign of Doppler shift. However, our current trained model is limited to the angle of view of the training dataset. Therefore, the next following step would be synthesizing 3D radar images, which shares the angle of view of our radar imaging systems, so that it can potentially enhance real radar images. Of course, due to the discrepancy between our training dataset of synthesized images and our test dataset of real images, the performance will degrade. We can either fundamentally compensate for this gap by improving the authenticity of synthesized radar images through 3D model fitting and specular reflection modeling, or we can try to include real radar images in the training set with ground truth obtained through synced vision detection outputs. Last but not least, in our current experiment, we only try to enhance images of cars, and we definitely want to gradually include more and more common participants of traffic. Eventually, we are looking forward to achieving comparable imaging quality as LiDAR while exceeding it



(a) detected cars [8] (b) input heat-map (c) ground truth (d) generated image

Figure 6: Some results from the second experiment. From top to bottom: 1) The car's boundary, shape and orientation has fully been recovered. The gray-scale colors do not much. However, colors are of little importance in our application. 2) The orientation of the middle car is reversed. As we shall see in the next section, 180 degrees of ambiguity can easily be fixed using the Doppler effect. 3) Although there is high reflection from the middle car, its boundary has correctly been identified as a small distant car. 4) Same as 3. Also the orientation of car on the side has been correctly estimated. 5) The generated image does not make sense. This is most likely because the sides of cars were rarely included in the training images provided to the network.

in reliability and velocity measurement, so that Millimeter-wave systems can play a deterministic role in both perception and connectivity of the vision of autonomous and smart transportation.

References

[1] Altair feko. 3

[2] Automated vehicles for safety. 1

[3] The cityscapes dataset. 2, 3, 4

[4] S. Chen and H. Wang. Sar target recognition based on deep learning. *International Conference on Data Science and Advanced Analytics, DSAA*, 2014. 2

[5] S. Fidler, S. Dickinson, and R. Urtasun. 3d object detection and viewpoint estimation with a deformable 3d cuboid model. *Conference on Neural Information Processing Systems, NIPS*, 2012. 4, 5

[6] M. Gong, J. Zhao, J. Liu, Q. Miao, and L. Jiao. Change detection in synthetic aperture radar images based on deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems, DSAA*, 2016. 2

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[8] K. He, G. Gkioxari, P. Dollr, and R. Girshick. Mask r-cnn. *arXiv:1703.06870 [cs.CV]*, 2017. 4, 5, 7

[9] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017. 2, 3

[10] W. Jones. Keeping cars from crashing. *IEEE Spectrum*, 2001. 1

[11] B. Mamandipoor, G. Malysa, A. Arbabian, U. Madhow, and K. Noujeim. 60 ghz synthetic aperture radar for short-range imaging: Theory and experiments. *Asilomar Conference on Signals, Systems and Computers*, 2014. 1

[12] A. Nambi, S. Bannur, I. Mehta, H. Kalra, A. Virmani, V. Padmandabhan, R. Bhandari, and B. Raman. Demo: Hams: Driver and driving monitoring using a smartphone. *ACM MobiCom*, 2018. 2

[13] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016. 2

[14] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3

[15] C. Schwegmann, W. Kleynhans, B. Salmon, L. Mdakane, and R. Meyer. Very deep learning for ship discrimination in synthetic aperture radar imagery. *IEEE International Geoscience and Remote Sensing Symposium, IGARSS*, 2016. 2

[16] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *arXiv preprint arXiv:1711.11585*, 2017. 2

[17] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shape modeling. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2015. 4

[18] H. Zhang, V. Sindagi, and V. M. Patel. Image de-raining using a conditional generative adversarial network. *arXiv preprint arXiv:1701.05957*, 2017. 2

[19] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint*, 2017. 2

[20] T. Zhang, A. Wiliem, S. Yang, and B. Lovell. Tv-gan: Generative adversarial network based thermal to visible face recognition. In *2018 International Conference on Biometrics (ICB)*, pages 174–181. IEEE, 2018. 2

[21] M. Zhao, T. Li, M. Alsheikh, Y. Tian, H. Zhao, A. Torralba, and D. Katabi. Trhough-wall human pose estimation using radio signals. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2018. 2

[22] M. Zhao, Y. Tian, H. Zhao, M. Alsheikh, T. Li, R. Hristov, Z. Kabelac, D. Katabi, and A. Torralba. Rf-based 3d skeletons. *SIGCOMM*, 2018. 2