Improving Millimeter Wave Radar Perception with Deep Learning

Junfeng Guan Sohrab Madani
ECE 544 Project Reportl
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 make correct decisions. As the most fundamental task, previous works have demonstrated accurate object detection and classification, but they are limited to data obtained from LiDARs and cameras. These optical sensors have high imaging resolution, but they naturally fail in low visibility conditions such as fog, rain, and snow, because light beams are narrower than water droplets and snowflakes.[?] This fundamental limitation of optical sensors is one of the major roadblocks to achieving the 5th SAE level of full automation. [?]

On the contrary, Radar has desirable propagation characteristics through small particles and can provide an alternate imaging solution in such inclement weather. Besides, radar can also directly measure the velocity of objects with the doppler shift of reflected signal without going through cluster tracking across frames. Although the low resolution of traditional automotive radar overshadows its advantages, the advent of Millimeter-wave antenna array technology provides an opportunity to have a reliable imaging system in inclement weather with higher resolution at the same time. Along with good propagation characteristics, it also provides huge bandwidth and large-aperture antenna arrays. This enables accurate Time-of-Flight (ToF) and Angle-of-Arrival (AoA) estimation for imaging. However, we would like to push the resolution to be comparable to that of LiDAR which has been proved to be relibale, but the imaging resolution in Millimeter-Wave is still not high enough to allow for applications like object detection or scene-understanding. Moreover, with RF one faces the issue of specularity, where reflections from objects may not come back to the receiver depending on the angle of incidence of the transmitted signal. Therefore, in this project, we propose to develop techniques that can enable high resolution imaging in low visibility conditions with RF signals. Our goal is to use deep learning models to enhance the low resolution images obtained from Millimeter wave radars, and enable various crucial vision applications for autonomous vehicles like lane detection, image mapping, localization, and object identification.

2. Challenges

1. The low resolution of range and angle estimation in radar imagaing appears as . Besides

model Problem with radar images is different and not well studied The low resolution specularity

related work classification RF-Pose3D [**?**] [**?**] demonstrates Convolutional Neural Network (CNN) that can recreate the human body skeletons by tracking 14 key points. Because CNNs leverage local dependencies in the data, it significantly reduces the total number of weights to be learned. Considering this favorable property of CNN, we are going to implement this model first in our project. In contrast, we will concentrate more on classifying features of vehicles to infer the shape, orientation, and even velocity. We are planning to start with 2D images, which contains the distance and angle of objects within a horizontal plane, then we will try to extend to 3D images.

New application

Presents an architecture htat leverages deep learning to sense using RF signals. Our architecture consists of a component that generates training example, a , and a sensing component that infers properties. We show how to build these components using

2. data Once the network is setup, it needs training data -i.e., it needs many labeled exp Dataset Variation between systems Experiment Processing

3.3D 3D CNN 3D GAN size complexity

4. Evaluation

3. Method

Describe the overall method on how you solve the proposed problem, and a bit of original derivation that has some relevance to what you're trying to accomplish

Problem statement: low blured images no boundary can be seen, specularity causes missing parts, no availbale dataset, 4D CNN NN complex.

Overview 3D, in the method, we say that we start with 2D version, and based on that build 3D. We are trying to use cGAN to generate higher resolution images from low resolution radar images, which should have a sharp and accurate boundary of the object. Also, the missing parts due to specualarity of reflection need to be feel up. S

Generative Adverserial Networks (or GANs) have been widely used to generate images [cite some stuff here] and fill in missing parts of data. 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 [1], 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 function for restoring images render blurry images, 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 adapted the CGAN from [cite pix2pix] 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

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

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. At the same time, 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

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

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.

As the generator, the authors of [pix2pix] used U-net [cite unet]. Similar to most architectures, U-net consists of a contracting path and an expansive path. The contracting path is made of two successive 3×3 convolutions 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. For our implementation, we chose n to be 70 where the image size was 256×256. This structure has been dubbed patchGAN [pix2pix].

Input: Low resolution radar images, because there is no available public dataset of mmWave radar images, and collecting a big enough dataset by ourselves is not possible. We synthesized radar images. This heat-map is fed into the network. Groundtruth: 1. Size of 3D which makes the training phase very slow even when using GPUs. 2. 2D: 3. 3D

The input to our problem is pre-processed radar data. First, using the raw data from the antenna array, a coarse heat-map of objects are generated. After some further processing.

Why not raw data: (Goes to detail of Dataset jayden) Radar imaging processing algorithms is a well establish feild for 70 years. and there is fruitless to try and learn them using machine learning. So instead of The reason why we did not choose the raw data as input is twofold.

3.1. Conditional GAN

3.2. Dataset Generation

Once we have designed our Unlike the digital camera, imaging radar less common, no dataset available . Since there is no

Challenge of unavailable radar dataset.

Experiment to collect radar images size time umbiguity

Simulation with EM

3D

2D Mask R-CNN input: radar image groundtruth: mask pros: large dataset with car truck human cons: No 3D info, no specularity 3D CAD input: contour groundtruth: pros: small dataset, single element cons: 3D shape info, specularity

Evaluate the simulation with EM simulation Feko and experimental results.

4. Experimental Results

Describe the setup of the experiments you ran, e.g., what evaluation metrics, datasets are used. Present the results, preferably in the form of tables and/or figures

4.1. Dataset

4.2. Results

5. Discussion and Conclusion

Analyze the results, summarize the findings and point out possible future directions

References

[1] I. M. H. K. A. V. V. P. R. B. A. U. Nambi, S. Bannur and B. Raman. Demo: Hams: Driver and driving monitoring using a smartphone. *ACM MobiCom*, 2018. 2