mmWave FMCW Radar-based Micro-Doppler Signature Feature Extraction and Classification

Edwin Pan
epan2@illinois.edu

Jingning Tang
jtang10@illinois.edu

Abstract

Recent advances in millimeter wave (mmWave) radar technology has allowed radar sensors to become a viable sensor complimentary to vision-based sensing modalities. With the form factor reduction and near-imaging capability, radar can shine in application with low-light and extreme weather scenarios. However, many research endeavors simplify the radar data as one detection point per moving object, neglecting the powerful aspects of imaging radar's capabilities. Micro-Doppler motion, the micro-motion of the moving object extracted from the signals' phase shift, is one such neglected capability. It can serve as an unique signature of the object or a type of movement.

In this work, we explored the extraction of micro-Doppler signatures from moving objects with a Texas Instruments 1843 frequency-modulated-continuous-wave (FMCW) multiple-input-multiple-output (MIMO) millimeter wave radar sensor. Also, using simulated data from MATLAB's^{TM}[7] Phased Array Toolbox, we investigated possible classification of targets based on the short-term Fourier transform (STFT) of the micro-Doppler signature.

1. Introduction

As the form factor of the millimeter wave (mmWave) radar hardware decreases drastically, radar is growing into a competitive sensing platform for complementing other sensing modalities such as camera and LiDAR. Compared to camera, radar is immune to bad weather and able to detect objects with minimal intrusion of people's privacy. Radar is also naturally sensitive to phase shift and therefore is superior in detecting the speed and micro-motion of the target. The micro-motion, commonly referred as micro-Doppler signature, of the target can be regarded as unique, and it provides additional information that is complementary to existing methods for target classification and recognition[10]. For our final project in Machine Learning in Signal Processing (CS598 PS), we propose to extract micro-Doppler signature from multiple objects, learn how to classify them in a radar-based solution.

This is a continuation of a project started within the Alchemy program. These projects span a total of 18-months and aim to bring deep technology to commercial viability. Our current pipeline is called OpenRadar[8], and performs all radar signal processing tasks from ADC data input to object tracking with Extended Kalman Filter (EKF). Some of the more common core low-level DSP code we had previously developed were reused. This provided us a solid foundation to explore more complex machine learning and signal processing problems. The tasks are split into two parts:

1. Extraction of the micro-Doppler signature given multiple objects are on the scene.

2. Given the spectrum of the micro-Doppler signature, explore the effectiveness of different classification algorithms including Gaussian Mixture Model (GMM) and Convolutional Neural Network (CNN).

The rest of this report is organized as follows. Section 2 describes the background and current efforts regarding to micro-Doppler feature extraction and classification. Section 3 presents the multi-object feature extraction. Section 4 shows the classification. Section 5 talks about the contribution and concludes the paper.

2. Background

2.1. Research Effort

In the world of radar sensing, mechanical vibration and rotation of structures in a target may induce frequency modulation on returned signals and generate side-bands about the center frequency of the target's body Doppler frequency[2]. The modulation of vibration and rotation, which individually generate lower and higher frequency relative to Doppler frequency, is described as the micro-Doppler. Since each type of the movement is distinct in terms of the rotation and vibration of objects' different parts,the micro-Doppler can be used as a distinct feature to classify different objects. For example, Eeden *et al.*

[9] uses micro-Doppler to differentiate human from animals in the wild, whereas Fioranelli *et al.* [4] applies the micro-Doppler-based human movement classification in the healthcare domain. To properly utilize the micro-Doppler feature, researchers often use the STFT to convert the samples into the micro-Doppler plot, then either model it as time-series, or convert it to image classification problem. Typical machine learning algorithms are Hidden Markov Model (HMM), Support Vector Machine (SVM), or Convolutional Neural Network (CNN).

2.2. Derivation of micro-Doppler

To better understand how the micro-Doppler is used in the classification problem, the derivation of micro-Doppler shall be briefly described. Figure 2 depicts a typical signal processing chain from a FMCW MIMO radar. The transmitter will emit frequency-modulated wave (usually called chirp) consecutively for multiple times each round (one round is usually called one frame). The input data is then gathered from all the receivers, demodulated with the base frequency, and thus is a matrix with the shape of numChirpsPerFrame × numReceivers × numADCSamples. Each chirp will be reflected to the receiver at different times due to objects being located at different ranges. Therefore, received signals will consist of different frequencies. Taking FFT on the direction of numADCSamples will reveal the frequencies and produce the ranges of all the detected objects.

As for the velocity of an object, two consecutive chirps, once encountering an object, will have different phases because of Doppler effect. Following the similar fashion, FFT along the numChirpsPerFrame dimension will provide the velocity information. Therefore, at each time step, a 2D FFT on the input data will provide a complete picture of the range-Doppler information under the current radar scene. If slicing into a certain range, it will reveal the micro-Doppler of all the objects at this time step, and if the data is gathered through multiple steps, the complete micro-Doppler can be computed.

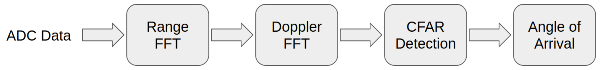


Figure 1: Typical Processing Chain of a FMCW Radar

From Figure 1, our microdoppler processing chain creates a fork in the typical processing pipeline right after the Doppler FFT, and saves the range-doppler plots over time. From here, naive threshold based noise removal and filtering is done on the collective range-doppler plots before being stitched together into a microdoppler spectrogram. The results of this forked pipeline can be viewed in Figure 2.

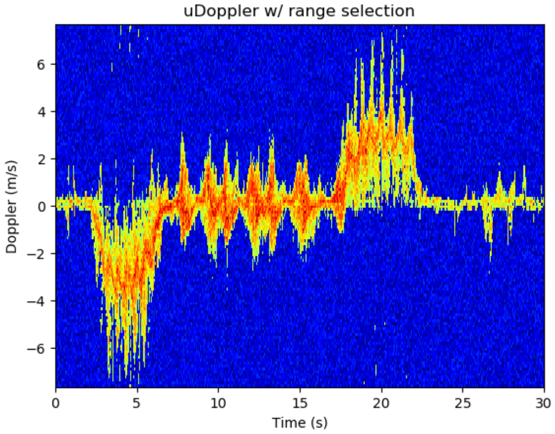


Figure 2: Micro-doppler spectrogram generated using pipeline written for this exploration. (2-6 sec): person walking toward radar. (6-17 sec): person squatting in place. (17-22 sec): person walking away from radar. (26-28 sec): person turned around to ask for instructions.

From Figure 2, we can see that the microdoppler plot can capture visually different patterns depending on the object's movement. However, if there are multiple objects present in the scene, such the microdoppler plot alone would not differentiate them. The single target assumption is one that we would like to eliminate. Distinguishing between the micro-Doppler signatures of multiple objects is one of the two main areas we build on in this exploration. Details will be discussed in the section 3.

2.3. Classification of micro-Doppler Signature

The micro-Doppler signature spectrogram classification problem will be addressed using two different classification techniques; one supervised, the other unsupervised.

The first proposed technique is to learn multiple multivariate Gaussian distributions based on the features and weights obtained by performing Principle Component Analysis (PCA) via Singular Value Decomposition (SVD). Given an $M \times N$ input data set X with N samples each M dimensions, SVD decomposes X as follows,

$$X = U\Sigma V^T$$

which maps X into an orthonormal eigenspace, scales it by the singular values, and back again. By choosing the ordering of the singular values to be from most significant to least significant, we choose to only keep some k bases, thereby reducing the dimensionality of the original data set X. The mapping of the eigenspace defined by the orthonormal bases to X is the weights matrix that we use to learn a

k dimensional Gaussian distribution defined below.

$$P(x;\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

The second proposed technique is to define a Convolutional Neural Network (CNN) to learn the distinguishing characteristics between the two spectrogram plots. A CNN is a specialized neural network layer architecture layer designed to work well with data that contains spatially local significance. Each convolutional layer performs a weighted sum of a local region of data points from the previous layer. The weighted sum is defined by the convolutional kernel used in that specific layer. Convolutional layers are generally followed by pooling layers, which downsample the features. This is generally sensible for images, since most significant changes occur at the edges rather than throughout the body of an object.

For the purposes of this exploration, the classification is treated as an image classification problem rather than a frequency varying time series classification problem. Treating this problem space as a time series classification issue makes sense if each class could change from one to the other over time. Indeed, this is true. A pedestrian is fully capable of transitioning into riding a bicycle. Additionally, a bicyclist is fully capable of getting off the bike and becoming a pedestrian. Consequently, a fully deployable application of this exploration would indeed need to take this into account. However, our current governing assumption is that the probability of transition between classes is negligible. This assumption has two motivating factors.

1. Addressing the transition between pedestrian and bicycle is an edge case that, while necessary to address, would significantly complicate the data capture and labeling process.

2. Since a vast majority of scanned targets should be strictly pedestrian or bicycle, any accuracy in classifying each class should not change significantly once adding the possibility of transitioning.

3. Extraction

In the section 2.2, it was briefly mentioned that most current researches assume that there is only one object captured by radar. However, the real-world applications, such as autonomous driving tasks, require that radar can properly operate when there are pedestrians, bicyclist and cars existing simultaneously. To achieve this capability, it is necessary to perform source separation on the input signals to generate n micro-Doppler spectrograms for all n detected objects.

Since micro-Doppler is simply the range-Doppler over time, the highlighted areas on the range-Doppler over time will be the micro-Doppler of all the objects, and therefore a natural induction to the solution would be performing clustering on the range-Doppler plot per frame. To exclude noises such as multi-path reflection or ground bounce, Density-based spatial clustering of applications with noise[3] (DBSCAN) is usually preferred. This method, however, would bring two problems. First, the clustering on range-Doppler will have difficult time identifying objects that are moving at the same speed and are equidistant from the radar. Second, while the clustering might capture each object's micro-Doppler, it will not associate the signatures to the objects accross multiple frames.

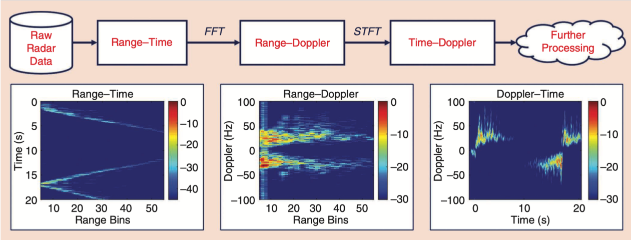


Figure 3: Range-Doppler to Micro-Doppler[4]

Since the modern state-of-the-art radar commercial radar systems are majority MIMO devices (the TI 1843 included), we can use the multiple receivers to generate delays and calculate angles of arrival along the azimuth dimension. Thus, to overcome the first shortcoming, one can switch to using range-azimuth information and perform the DBSCAN clustering. Objects at different locations will not overlap as long as the azimuth resolution is high enough and DBSCAN parameters are tuned properly. However, radar poses a new issue in this domain. Due to the conical shape of the transmitter emission patterns, radar samples are sparser as the detected objects move farther away. To perform clustering accurately, DBSCAN has to be modified to take the dynamic sparsity into account. As for the second problem, one can jointly separate the micro-Doppler and perform the multi-object tracking with Kalman Filter. The details of these two solutions shall be discussed below.

3.1. DBSCAN

The general idea of DBSCAN is to search the neighborhood of a point to check if the density of points of such neighborhood is beyond the criterion. The size of the search area and the density criterion are tunable parameters from the algorithm. Since these two parameters are fixed across all the data points, it poses a problem for non-equidistant data samples, such as radar's data. As the object moves away from the radar, the radar will produce fewer detections. For example, For a car at 3 meters, 59 azimuth cells are affected compared to only 3 cells for a car at 80 meters[5]. As a result, not only should DBSCAN require fewer points to be qualified as a cluster, but the search grid should also be reduced for the farther objects. There are variable density algorithms invented already, such as Ordering Points To Identify the Clustering Structure (OPTICS) [1], but our endeavor will focus a relatively simply yet effective method called Grid-Based DBSCAN, proposed by Ester *et al.* [5].

The improvement upon original DBSCAN is quite straightforward. Since it requires different search criteria based on different locations, the criteria is simply calculated for each search point. First the ratio between radial and angular distance ($c_{i,j}$) is calculate based on

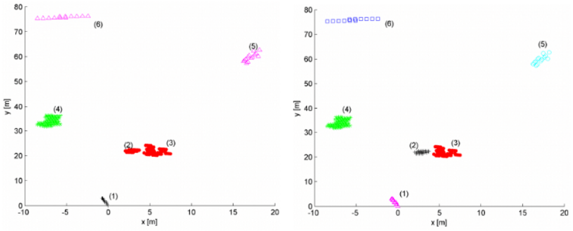
$$c_{i,j} = \frac{r_{i,j}}{2\Delta r}\big(\sin(\theta_{i,j+1} - \theta_{i,j}) + \sin(\theta_{i,j} - \theta_{i,j-1})\big)$$

where ($r_{i,j}$ and ($\theta_{i,j}$ is the radial and angular distance at the grid (i, j) and Δr is the radial resolution. Then based on the ratio, the search-distance along the θ-distance is calculated as the follow.

$$w_{i,j}(c) = \frac{1}{f \cdot c_{i,j}}$$

where f is a parameter to set the shape of the search area. The search area will be a circle if $f = 1$ and ellipse when $f \neq 1$.

Due to our limited ability to capture complex datasets with multiple objects moving in and out of frame in an electromagnetically clean environment (i.e. an anechoic chamber or large open area), only the algorithm itself is implemented. Using existing data we have captured previously, our Grid-Based DBSCAN has been tried on scans containing no more than three objects in the field of view. However, according to original paper, it improves significantly from the baseline DBSCAN. From above, the clusters from 1 to 6 are supposed to be noises, pedestrian, car, two other vehicles and a barrier. For the original DBSCAN, since the criterion of clustering is independent from the search grids, object 5 and 6 are naturally classified as noise and 2 and 3 are clustered together. As for the gird-based improvement, the algorithm correctly separates the pedestrian 2 and car 3, while figuring out 5 and 6 are actual object and 1 being noises.



(a) Original DBSCAN (b) Grid-based DBSCAN

Figure 4: A comparison of baseline DBSCAN and grid-based DBSCAN for clustering cars and pedestrian[5]

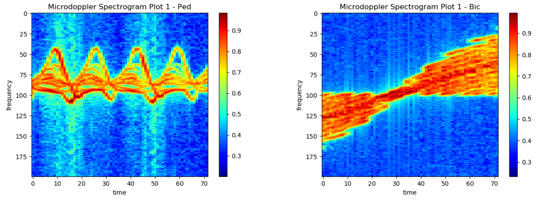
3.2. Joint Feature Extraction and Tracking

To continuously extract one object's micro-Doppler, the data points associated to the object need to be tracked constantly for each time frame, the task of which cannot be finished simply by DBSCAN. Given the objects are moving continuously, one can simply associate clusters across two frames to achieve the tracking by comparing the mean position and velocity of the clusters and the overlapping of clusters between two frames.

Kalman Filter is a superior way to achieve the constant tracking of the object, by combining the modeling of objects' movement and actual measurement. While we have a working implementation of the Extended Kalman Filter (EKF), we spent so much time on refactoring the code for the software purpose that we didn't finish integrate it into the current system by the time of submitting this report.

4. Classification

With the ability to obtain isolated micro-Doppler scans, the next step is to perform classification on the isolated micro-Doppler spectrograms. In real world roadway scenarios, there are a vast array of different objects a car may see. To make the problem simpler, we chose two common roadway objects that are relatively difficult for cameras to discern; pedestrians and bicycles. Thus, the main objective in this exploration is to distinguish between humans that are walking (pedestrians) and humans that are riding a bike (bicyclists) using micro-doppler spectrogams with reasonable accuracy.



(a) Pedestrian spectrogram (b) Bicycle spectrogram

Figure 5: A visual comparison between pedestrian and bicycle micro-doppler signatures via STFT spectrogram

4.1. Data Set

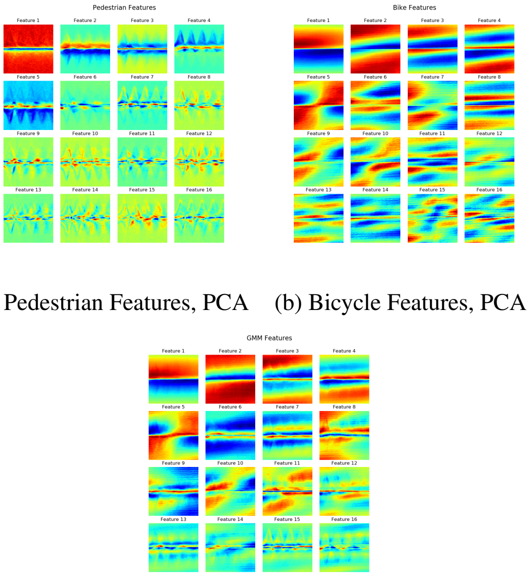
Due to limitations in our capacity to gather data, we've supplemented the training data sets gathered in our previous Alchemy work with data that has been simulated using the MATLAB^{TM}R2019 Phased Array Toolbox. In total, we used 9586 total microdoppler spectrograms. Each spectrogram was limited to a 2 second scan time, with velocities up to ± 10 meters per second. The split between pedestrian and bicyclist scans was 51.2% (4809 spectrograms) and 49.8% (4777 spectrograms) respectively.

4.2. Pre-processing

Each spectrogram contains 400 time bins, and 144 frequency bins. Since our exploration considers these images, these bins can be thought of as 57600 pixel images (though each pixel is a single precision floating point rather than an 8-bit integer). To reduce this to a more managable size, the spectrograms were each downsampled by a factor 2 along both the time and velocity axis, bringing the total image shape down to 200 by 72 and 14400 pixels. The downsampling was done using a 2D running average filter. The running average filter is implemented by convolving each spectrogram with a 2×2 kernel with each kernel element being the value $\frac{1}{4}$. This decimation method was chosen over dropping every other pixel because of the running average filter's comparative robustness to aliasing.

4.3. Gaussian Mixture Model Classification

Taking the downsampled data, the 9586 spectrograms are vectorized, then split into training and testing. For this classification method, 2000 spectrograms were used for training, with the remaining spectrograms used for testing. The 2000 training spectrograms were split 50-50 between pedestrian and bicyclist plots. These then mean-centered. Next, they're run through PCA to produce 16 features and each spectrogram's corresponding weights. From here, two 16 dimensional multivariate Gaussian distributions were estimated based on the mean and standard deviation of the weights obtained through PCA.



(a) Pedestrian Features, PCA (b) Bicycle Features, PCA

(c) Combined Features, PCA

Figure 6: PCA features for different data sets

Having estimated two Gaussian distributions, a linear transformation is performed on each of the vectorized test spectrograms, bringing them into the orthogonal eigenspace calculated during PCA on the training set. The linear transform is calculated as follows:

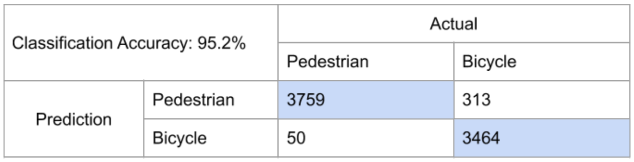
$$Z = W^T X$$

where W is the feature matrix, X is the vectorized test data, and Z is the resulting weights matrix. Using the weight vector for each test spectrogram, we locate the probability density at that point using both multivariate Gaussian distributions from training. The spectrogram will be classified as whichever Gaussian produced a higher probability.

Using this classification method, we were able to achieve 95.2% accuracy on the unseen test set. We believe there are two major reasons for this high accuracy.

1. The data samples used to train are incredibly clean and devoid of significant noise. A typical radar scan contains by both zero-mean white noise and multi-path reflections (similar to an audio echo). The two sources of data used in this classification problem are the sole reason for the clean plots. Naturally, simulated data demonstrates an ideal scenario which is inherently clean. Additionally, source separation is a good method for removing multi-path reflections, which often come in at angles that differ from the true target.

2. The two classes exhibit significantly different motions in their actions, leading to distinctions that are readily distinguishable via PCA. Looking at Figure 6(b), it's immediately clear that bicyclists have motion that manifests on the spectrogram as broad, high velocity and low spacial frequency features (referring to frequency in terms of image pixels). On the other hand, Figure 6(a)'s pedestrian features are a lot lower velocity and higher frequency. When features are extracted from their combined data set, the first few features appear very similar to the features found from the bicycle feature set. Additionally, the later features from the combined set resemble those from the pedestrian feature set. Even within the feature set itself, one can already begin to distinguish between bicycles and pedestrians.



Classification Accuracy: 95.2%		Actual	
		Pedestrian	Bicycle
Prediction	Pedestrian	3759	313
	Bicycle	50	3464

Figure 7: Learned Gaussian mixture model based classification confusion matrix

4.4. Convolutional Neural Network Classification

When reviewing MATLAB$^{\text{TM}}$'s data generation scripts, we came across a demonstration where they built a Convolutional Neural Network in the MATLAB$^{\text{TM}}$language to distinguish between 5 classes: pedestrian, bicycle, 2 pedestrians, 2 bicycles, 1 bicycle and 1 pedestrian. Seeing our own pipeline as a means of demonstrating that we can achieve similar results via source separation, we went ahead with building a CNN that could hopefully beat the accuracy bar set by the GMM classification process. Plus, CNN's are responsible for some of the best results in image processing. Since we had decided early on to treat the classification problem as an image processing problem, this solution fit the mold.

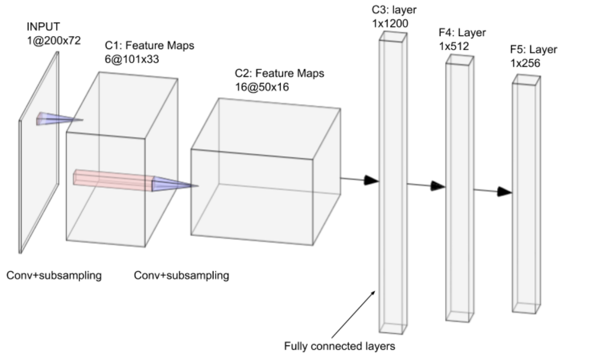
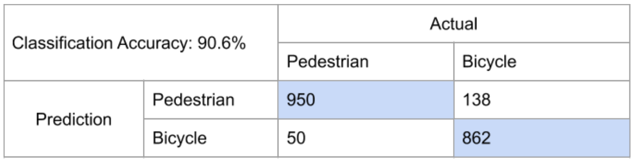


Figure 8: CNN Architecture used for this exploration

The CNN architecture is as follows. First, 6 features are extracted in the first convolutional layer, with rectangular kernels of shape 3×7. This is batch normalized, and 2×2 max-pooled. The next convolutional layer produces 16 features with a 5×5 kernel, batch normalized, and 2×2 max-pooled again. Three fully connected linear layers are added at the end, before the CNN outputs two values for it's prediction. The argmax of those values is the class the model predicts.

A few key points were kept in mind during the design of this CNN's architecture. First off, CNNs excel at image processing because they limit their features to deal with spatially local regions of the input image. In fact, each local region has no impact on the feature vectors of the others until the fully connected layers at the very end. While this is very useful for typical images, which exhibit significant local correlation (with most significant changes happening at the edges of objects), our problem constituted the use of spectrograms as images. To be clear, we weren't certain that there wasn't any meaningful relationship between the pixels that appeared spatially far way from each other. To work around this, we defined the first layer's kernel to be rectangular, including more points in the velocity direction than the time direction. This was decided because all the velocity bins at each time step come from the same scan. Moving over one time bin equates to moving over one entire scan. Second, the SVD calculation for PCA was computationally expensive for the task we had given it. Thus 16 features (while successful), could potentially be a limiting factor. Using the CNN's iterative convergence on a solution (and PyTorch's easy utilization of GPU processing), we could hopefully break the 95% benchmark set by the unsupervised method.

Using the same preprocessed data from the GMM classification scheme, we swapped the training and testing sets so as to allow for more training samples. However, despite extensive tuning of hyper-parameters, we were unable to beat the GMM based classification accuracy. As noted earlier, CNNs force an assumption on what data within the plot is related to other parts of the plot. While this can certainly be true for a typical image, spectrogram plots contain more nuanced relationships that are actually ignored by moving to a more complex classification setup like a CNN.



Classification Accuracy: 90.6%		Actual	
		Pedestrian	Bicycle
Prediction	Pedestrian	950	138
	Bicycle	50	862

Figure 9: Convolutional Neural Network based classification confusion matrix

What we aimed to achieve with the CNN wasn't a means of improving the GMM's results. Rather, it was an alternative means of arriving at the same result with a more complicated setup. In hindsight, it might have been more worthwhile to capture and label the data necessary to augment the Gaussian mixture model based classification with some time series components, and build up a Hidden Markov Model approach to solving this classification problem.

5. Conclusion

Through the exploration of micro-Doppler, we have learned how powerful the signature is in order to help classify objects. Knowing that most industry solutions still put single point on each detected object, leaving the micro-Doppler unused, a complete solution with a high-end radar can be either a great research breakthrough or an exciting industry solution. As for the improvement upon the explored solution, one can try adding Doppler information in the DBSCAN, or deep learning-based method to replace the DBSCAN, given more data or an easy setup to consistently capturing data on the road.

6. Verdict

Edwin Pan is in charge of exploring classification on the micro-Doppler data while Jingning Tang put his effort on the multi-object feature extraction. The feature extraction part is not finished by the end of the class, leaving the implementation and integration of Kalman Filter in the future effort. The following ideas can be pursued to improve the completeness of the project:

1. Follow a well-explained paper with provided data to reproduce the work. Steering the effor from the end-to-end feature-extraction and classifying to spliting the work was a good idea, but we can also follow the Google Soli's work [6].

2. Trail-blazing was much more interesting than just learn a network on the given dataset. However, it means losing focus was quite easy. In the process, Jingning spent too much time on understanding the noise in the DSP process and the implementation of the Kalman Filter, which led to the tracking module unfinished.

References

[1] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD '99, pages 49–60, New York, NY, USA, 1999. ACM.

[2] V. C. Chen. Analysis of radar micro-doppler with time-frequency transform. In *Proceedings of the Tenth IEEE Workshop on Statistical Signal and Array Processing (Cat. No.00TH8496)*, pages 463–466, Aug 2000.

[3] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[4] F. Fioranelli, J. Le Kernec, and S. A. Shah. Radar for health care: Recognizing human activities and monitoring vital signs. *IEEE Potentials*, 38(4):16–23, July 2019.

[5] D. Kellner, J. Klappstein, and K. Dietmayer. Grid-based dbscan for clustering extended objects in radar data. In *2012 IEEE Intelligent Vehicles Symposium*, pages 365–370, June 2012.

[6] J. Lien, N. Gillian, M. E. Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, and I. Poupyrev. Soli: Ubiquitous gesture sensing with millimeter wave radar. *ACM Trans. Graph.*, 35(4):142:1–142:19, July 2016.

[7] MATLAB. Pedestrian and bicyclist classification using deep learning. `https://tinyurl.com/ub8mjl7`, 2019.

[8] PreSense. Openradar mmwave package. `https://github.com/presenseradar/openradar`, 2019.

[9] W. van Eeden, J. de Villiers, R. Berndt, W. Nel, and E. Blasch. Micro-doppler radar classification of humans and animals in an operational environment. *Expert Systems with Applications*, 102:1 – 11, 2018.

[10] J. Zabalza, C. Clemente, G. Di Caterina, Jinchang Ren, J. J. Soraghan, and S. Marshall. Robust pca micro-doppler classification using svm on embedded systems. *IEEE Transactions on Aerospace and Electronic Systems*, 50(3):2304–2310, July 2014.