Received May 7, 2021, accepted June 3, 2021, date of publication June 7, 2021, date of current version June 16, 2021.
Digital Object Identifier 10.1109/ACCESS.2021.3087207

Automatic Label Creation Framework for FMCW Radar Images Using Camera Data

JAVIER MENDEZ[1], STEPHAN SCHOENFELDT[1], XINYI TANG[1], (Member, IEEE), JAKOB VALTL[1], M. P. CUELLAR[2], AND DIEGO P. MORALES[3]
[1]Infineon Technologies AG, 85579 Neubiberg, Germany
[2]Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain
[3]Department of Electronics and Computer Technology, University of Granada, 18071 Granada, Spain

Corresponding author: Javier Mendez (javier.mendezgomez@infineon.com)

This work was supported in part by the Project A-SWARM through the German Federal Ministry of Economy and Industry (BMWI) by the Maritime Forschungsstrategie 2025 under Project 03SX485D.

ABSTRACT Data acquisition and treatment are key issues for any Deep Learning (DL) technique, especially in computer vision tasks. A big effort must be done for the creation of labeled datasets, due to the time this task requires and its complexity in cases where different sensors must be used. This is the case of radar imaging applications, where radar data are difficult to analyze and must be labeled manually. In this paper, a semi-automatic framework to generate labels for range Doppler maps (radar images) is proposed. This technique is based on a sensor fusion approach with radar and camera sensors. The proposed scheme operates in two steps: The first step is the environment features extraction, in which the radar data is preprocessed and filtered to remove ghost targets and detect clusters, and camera data are used to extract the information of the targets. In the second step, a rule-based system that considers the extracted features fuses the information to generate labels for the radar data. By using the proposed framework, the experimentation performed suggests that the time required to label the data is reduced as well as the possibility of human error during the labeling task. Our results show that the proposed technique can improve the final model accuracy with regards the traditional labeling method, carried out by human experts.

INDEX TERMS Sensor fusion, machine learning algorithms, deep learning, radar, auto-labeling system.

I. INTRODUCTION

In recent years, radar imaging techniques have been proven to provide high performance results when used for classification tasks in autonomous driving [1], [2], object detection [3], [4] and activity recognition [5], [6]. Researchers have also studied the integration of Machine Learning (ML) techniques for the radar signal preprocessing [5] as well as the previously commented tasks [2], [3], [6] to achieve high performance results. However, large application-specific datasets are required when training Deep Learning (DL) models for these purposes using a supervised approach.

The creation of new datasets is a current problem due to the required time to gather the data and, especially, to correctly label the data. In order to solve this problem, new training approaches are been researched to avoid the labeling step of the dataset. An example of this is the use of Reinforcement Learning [7], [8], where the DL model is trained without a dataset itself but in a simulated environment. Even when these techniques are being researched, a large set of applications still use supervised training [9], [10]. Therefore, labeled datasets are still required for new applications.

In recent approaches, some authors have studied the implementation of ML techniques to label the datasets automatically. These techniques aim to reduce the human errors in the labeling process as well as the time required for this task. Some examples of these techniques are proposed in [11]–[15], which are deeply analyzed in Section 2. Some of these techniques are based on a Sensor Fusion approach such as [1]. These Sensor Fusion techniques can be divided into Early Fusion and Late Fusion pipelines. The Early Fusion approach combines the data with a low level of preprocessing to generate new raw data that can be later studied as a single input. The Early Fusion can also be used to extract the final information from the initial raw data from multiple sensors. This technique is beneficial when the data can be merged easily due to similarities in the format or the features. The issues of this technique fall into the restriction of using data

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Tan.

with similar formats or implementing complex algorithms to overcome these differences as in [1]. On the other hand, the Late Fusion combines the data at a later step after a deeper preprocessing of the data. This approach can be used to efficiently study each data separately to determine what features can be useful before the merging step. This leads to a further preprocessing that can be optimized for each data type. At the same time, the complexity of the fusing algorithm can be reduced.

Conventionally, manually labeling radar data is performed frame by frame. The efficiency of this process is limited due to the fact that a ground truth must be provided for each radar image. At the same time, it is possible to incur in human errors during the execution of this task, mainly as misinterpretation of the data. This is a result of the complexity and non-intuitive visualization of the radar data.

The developed framework in this paper aims to help during the dataset creation using a Late Sensor Fusion pipeline. This pipeline, based on DL models as well as traditional approaches, merges relevant features from input data after an individual preprocessing. These specific preprocessing techniques are optimized to extract accurate and relevant information from the camera and radar sensors. The developed framework aims to generate labels for the studied dataset. By applying this pipeline, the possibility of incurring in human error is reduced, and also the required time to study new data, as shown in the experiments in Section 4.

As a secondary objective, we study the implementation of the resulting DL models in edge devices with low hardware resources. For this reason, different Deep Learning frameworks have been researched as well as the memory allocation of the data. The experimental section discusses performance both in accuracy and time, respectively.

The remaining of the manuscript is organized as follows: Section 2 focuses on related works to the addressed problem. In Section 3, the Sensor Fusion pipeline is described for a deep understanding of the label creation process. After that, Section 4 will focus on the experiment where this framework has been applied, and finally Section 5 concludes.

II. RELATED WORKS

The use of radar sensors for classification tasks is increasing its popularity. This enables a system to recognize its environment [1]–[4] without having to deal with privacy concerns as it happens when using camera data. However, this leads to a requirement of large specific labeled datasets for each application (when using AI models that required a supervised training). The creation of labeled datasets is a bottleneck in the ML model creation due to the time required for its creation as well as the possibility to incur human errors during the labeling process. As a result, numerous researchers are working on this field.

An example of this approach is the object detection based on the fusion of these sensors researched by Nobis *et al.* [1]. This Early Fusion approach is based on the data fusion using an Artificial Neural Network (ANN). The raw data from the camera sensor and the low-preprocessed radar data are used as inputs of the ANN that execute the fusion and classification process. At the same time the system studies at what point the fusing process should be executed to obtain the best classification results. In this technique, the radar data is used as reinforcement, as 2D points, for the camera image in the ANN. This technique proved its utility in environments where the camera data are corrupted or they do not provide enough information, for example in dark environments or with extreme weather conditions such as rain. The accuracy provided with this approach exceeds the state-of-the-art results obtained only with camera data in the NuScenes dataset as well as the Technical University of Munich (TUM) dataset. The limitation of this technique relies in the complex structure of the Deep Neural Network (DNN) designed for the fusion tasks. However, due to the study of multiple frames, this technique can achieve state-of-the-art results. The accuracy of this algorithm is further compared with the proposed framework in Subsection 4.4. where it is presented how our tool achieves a 24.556% higher accuracy when studying single frames.

Lim *et al.* [16] proposed a similar pipeline of [1] to the previously commented. Their proposed pipeline of Early Fusion to combine radar and camera information for target detection is based on an ANN. This ANN has an input for the radar data and another one for the camera data. Therefore, the data can be studied independently in a first step to extract high level features before merging them. The main difference of this technique respect [1] is the data type used. They use range-azimuth radar images instead of 2D points. This approach allows the system to employ feature pyramid network structures. Since there are no public datasets with this radar data, they built their own dataset to evaluate their technique, achieving a 73.5%. Their low accuracy in comparison with other techniques may be due to the Early Fusion approach followed. This limits the evaluation of the preprocessing of the data before the fusion step, what can lead to efficiency problems when extracting the features. At the same time, the proposed pipeline contains 2 Single Shot Detectors. As a result, the complexity of the algorithm is higher than our proposed pipeline.

Ji and Prokhorove [17] proposed a method to locate and classify objects based on radar and camera sensors. This method can be divided in two steps: In the first step, the data are preprocessed to extract the possible target location from the radar data using a Kalman filter and, in the second step, these possible points are projected into the camera plane to locate the relevant areas of the camera image to study. Later, these areas of the camera images are analyzed using a Multilayer In-place Learning Network to classify the objects into a set of known categories. The overall accuracy obtained with this method is 96.8% in the dataset studied in the paper which contains 400 images. Because of the pipeline of this method, the time required to classify all the objects depends on the number of objects due to the fact that each target is fed into the DNN individually in a loop approach. This loop approach may lead to high latency in environments with high density of targets in contrast with other techniques such as Single Shoot Detection (SSD) or Faster R-CNN where the whole image is studied at the same time by the DNN.

Kocic *et al.* [18] shows a pipeline for sensor fusion in the field of autonomous driving. The presented pipeline has three steps. In the first step, the data collected by multiple sensors is preprocessed to represent the same environment: LiDAR and radar generate 3D point clouds and the camera provides RGB images. These two types of data are fed into two different DNNs, one for the 3D points and one for the RGB image to generate labels for the objects in the environment separately. These labels are later fed into another DNN to execute a high-level fusion. Apart from localization, the results from the high-level fusion can be used to generate an occupancy grid surround the ego vehicle. The benefits of this structure lie on the fact it avoids lossy input predictions while using simple DNNs already used in other applications. At the same time, these factors are also the limitations for this method. Because of integrating multiple DNN in the same system, the resource requirements of this approach are higher than other techniques developed in previously commented papers as well as the technique explain in our paper. These multiple DNNs may also lead to high latency even when the paper does not research the output frame rate this pipeline can generate.

Zhang *et al.* [19] follows a similar approach to the previous authors. They propose a Late Sensor Fusion pipeline where a millimeter-wave radar is used to extract the position and the speed of the obstacles. The data of the location of the obstacles is used to generate a region of interest for a deeper study in the data collected by a camera sensor, where Machine Learning techniques are applied to extract reliable 3D bounding boxes for the objects and track them. The results obtained with this technique are the 91.6% of accuracy on dataset used for their experiment. Because of its similarities, the limitations of this technique lie in the same facts as the techniques presented by Kocic *et al.* [18] and Ji and Prokhorove [17].

The system proposed in [2] also uses radar data to classify and locate objects. The technique applied to study the data in this paper is based on the intensity and decayed spectrum. The approach used in that paper is based on a DNN following a traditional approach to create an Object Detector where first the clusters, based on range-azimuth intensity map, are located before been fed into a Convolutional Neural Network (CNN) to classify them. The paper proves how this approach can achieve state-of-the-art accuracy for the object classification without depending on any other sensor due to the fact that they achieve an accuracy of 65.30% at an initial frame, being later further improved with accumulated frames. Our technique can provide results on real time with higher accuracy due to the use of Range Doppler Maps (RDM) images instead of the range-azimuth intensity map followed by the authors of this paper.

Focus on the autolabeling process, as in our paper, Winterling *et al.* [11] proposed a technique based on a CNN to automatically generate labels for occupancy grid maps generated from radar data. In [11], the initial data is preprocessed to obtain an occupancy grid where each cell contains the information of the probability of an object being at that position. These maps are fed into a CNN where the author had to manually label a set of data in each iteration to improve the accuracy achieved. Therefore, this is an iterative process that still requires human interaction during the labeling process to manually label the unsure data of the dataset, in contrast with the proposed framework in our paper.

The problem presented in the previous paper was also researched by Cicco *et al.* [13]. In their manuscript, the authors explained how the creation of labeled datasets for Deep Learning is extremely time consuming due to the volume of data require as well as the complexity to label correctly each frame. As a result, they proposed a technique to create datasets for new applications in the agriculture field based on synthetic data creation. The dataset created with this technique proved its quality by training a DNN using this dataset and studying its final accuracy over a manually labeled dataset. This technique, even when it reduces the human effort and time to create the dataset, does not target the problem researched in our paper due to the fact that it use synthetic data instead of automatizing the labeling process as we have researched in our paper.

Suchi *et al.* [14] also proposed a technique to create a semi-automatic labeled datasets of RGB images at pixel level. During the dataset creation, their tool requires data from a depth sensor apart from the RGB images. The followed approach is based on comparing the distance of each 3D point with the neighbors points by exploiting spatial shifts in the depth data to determine if it belong to an object. To overcome the problems due to the variation of the distance of the objects respect the recording system, they include a preprocessing to adapt the system to the range of the target. This approach is similar to our technique in the sense it is based on a sensor fusion paradigm for the label creation. However, since we are targeting radar data, this technique cannot be used due to the fact that we are missing the data that a depth sensor could provide to accurately differentiate targets that are near each other.

Following the idea of combining data from multiple sources as in the previous technique, Meseguer-Brocal *et al.* [15] developed an approach to automatically create datasets of audio, lyrics and notes. Their technique is based on karaoke user data that contains annotations of time-aligned lyrics and notes. In a later step, this data is compared with audio candidates from the web to select the best candidate using a CNN. In the last step, a teacher-student paradigm is applied to improve the results obtained with this DL model.

Pursuing an implementation without using Deep Learning techniques, Tang and Lewis [12] researched the auto-labeling of images for image classification. They compared different algorithms for the labeling process on multiple datasets in order to establish the benefits from each technique. The techniques researched include techniques such as CSD-Prop, SvdCos, and CSD-SVM. Among the researched techniques, the CSD-SVM provided the best results taking into account the quantity of prior information required by the system. This technique labeled correctly 767 images out of the 4500 images in the Corel dataset [20] while the CSD-Prop and SvdCos labeled 577 and 349 images respectively. However, the number of correct labels does not achieve a large enough volume to be implemented as an automatic labeling system for the creation of labeled datasets.

After showing the most relevant research papers that justify our research, we can see that some of these papers explain how the use of radar in collaboration with ML techniques for object detection is an emerging research line. The accuracy as well as latency results of these techniques can still be further improved. This has led us to develop the semi-automatic labeling framework. In order to improve the previously explained techniques, some of the stages of the pipeline of our framework have been optimized to improve the latency as well as the memory consumption. At the same time, it is possible to see how previous automatic labeling techniques have not being able to create an accurate and efficient process in contrast with our proposed framework since their accuracy is not high enough [12] or they do not directly face the problem of labeling the data [13].

The approach proposed in our paper is based in an DNN to extract relevant features for the camera data. The later data fusion, at object level, is based on rule-based system approach using the cluster Angle of Arrival (AoA) and distance of the targets from each sensor to match the targets from both sensors. As a result, high accuracy is achieved in our dataset while maintaining low latency.

The proposed preprocessing method, as well as the Sensor Fusion approach, will be explained in detail in the following section.

III. SENSOR FUSION PIPELINE

This section describes the preprocessing techniques applied in the proposed framework to reduce the noise and reconstruct missing data. At the same time, this preprocessing extract high-level features to be fused in a later step. We remark that our final target application is self-autonomous driving and, more specifically, object recognition and localization.

The Sensor Fusion pipeline for the proposed autolabeling framework in this paper consists of two main modules: data preprocessing and data fusion. These modules can be subdivided depending on the source data sensor, as shown in the Figure 1.

This pipeline follows a natural information flow from raw data to feature extraction and a final step where these high-level features are fused to obtain the final data. The final data, extracted using this pipeline, is the information about the classification of the clusters from the RDM image, a bounding box estimation, the AoA and the presence of multiple objects in the cluster.

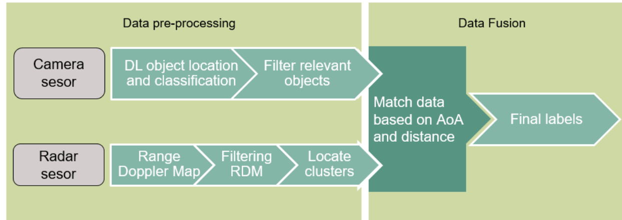


FIGURE 1. Sensor Fusion pipeline.

In order to extract relevant features, the data from both sensors is preprocessed separately due to the differences in their features/structures. As a result of this, a specific preprocessing pipeline is designed for each data type. The preprocessing of the radar data is based on a traditional signal preprocessing pipeline. The raw radar data is converted into a new format, RDM, where the relevant features can be extracted more efficiently. This is followed by a filtering step to reduce the noise. On the other hand, the camera data preprocessing is based on computer vision techniques, specifically DNN. DNN were chosen due to their high accuracy results achieved for computer vision in the literature. These preprocessing techniques are further explained in the next two subsections for a deeper understanding.

A. RADAR DATA PREPROCESSING

A Multiple-input-multiple-output (MIMO) frequency modulated continuous wave (FMCW) radar with four channels is used. The number of channels can be further increased by applying virtual array concept with a cost of lower frame rate. Calibration was applied to correct the AoA estimation errors after adding a radar random. The sampling rate, frame size, chirp bandwidth and chirp time are properly selected to ensure a good resolution within our region of interest.

In order to fuse the information from the radar data and the camera data, the object is first detected and located in the radar RDM data to reduce the data dimension. This technique transforms the time domain ADC signals to RDM images, where the information about the distance and speed of the targets is maintained. A Constant False Alarm Rate (CFAR) filter is then applied to the RDM image for target detection [21], [22]. This pipeline is presented in Figure 2.

The CFAR technique compares each bin of the RDM with its surrounding ones, setting a maximum range called training cells without including a subset of this set called guard cells. This filter is used to estimate the presence of a real target within the bin under test. There are training bins near the bin under test to compute the noise floor. Immediately adjacent bins to the bin under test are considered as guard bins and are ignored so that the possible leaked signals do not contribute to the noise floor computation. A bin is declared to contain an object when its value is greater than the scaled noise floor. With CFAR, the computed noise floor acts as a dynamic threshold instead of a fixed threshold value.

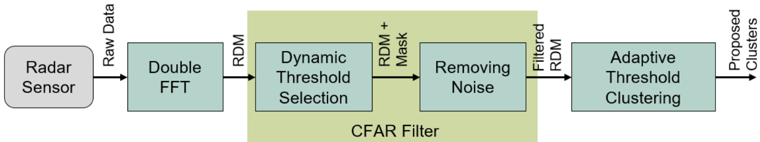


FIGURE 2. Radar data processing pipeline.

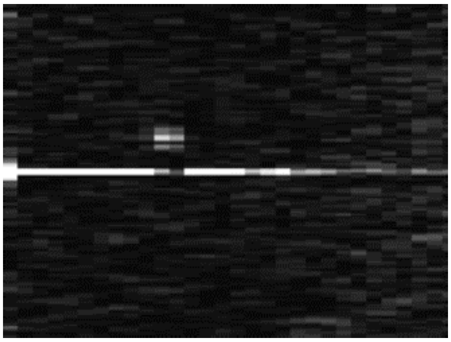


FIGURE 3. RDM frame example.

In Figure 2, the CFAR filtering has been divided into the two sub-processed required for its implementation: dynamic threshold selection and filtering the points whose value is under the threshold. This dynamic threshold is calculated using (1), where m and n are the horizontal and vertical number of training cells and $\hat{m}$ and $\hat{n}$ are the guard cells range horizontally and vertically.

$$T_i = (\sum_i \sum_j c_{ij})/(4(mn - \hat{m}\hat{n})) \quad i \in [-m, -\hat{m}] \cup [\hat{m}, m]$$
$$j \in [-n, -\hat{n}] \cup [\hat{n}, n] \qquad (1)$$

At this point, it is possible to extract clusters of real targets from the RDM images with high accuracy. An adaptive threshold technique has been used to determine the limits of the clusters. For each of the proposed clusters, a proportional threshold to its maximum value is generated. This threshold can be divided into speed threshold and range threshold to study the vertical and horizontal bins respectively. A later algorithm will use these thresholds to study the surrounding points of each cluster to determine the associated bounding box of each object. Therefore, the detection of a target is independent of target size (due to the precision of the radar sensor, real target size or distance to the object) in the RDM image.

It is important to understand that the white horizontal central line in the RDM images, as shown in Figure 3, represents the static objects captured by the radar sensor. In this representation, when multiple targets are static, they generate a line in the no speed area (the center line) that must be ignored since they cannot be separated for a proper classification.

These detected clusters will be later used in the fusing process with the targets located in the camera data to extract the label data expected from this framework.

B. CAMERA DATA PREPROCESSING

The data recorded from the camera will be fed into a DL object detector to extract the classification, AoA and location of the targets in each frame as it is presented in Figure 4.

The SSD structure [23] has been chosen due to its capabilities to enable real-time detection in comparison with other state-of-the-art structures such as Faster R-CNN [24]. SSD structure speeds up the processing of the data by removing the region proposal network present in other structures. To recover from the accuracy drop due to this, SSD applies techniques such as multi-scale features and default boxes. These improvements increase its accuracy to match the Faster R-CNN's accuracy while using lower resolution images. As a result, the size of this model as well as the resource requirements to execute it are reduced in comparison with other DL models for object detection.

The loss function of the model consists of two terms, the localization loss and the confidence loss. The first loss is the loss related to the position of the bounding box of the detected targets, penalizing only predictions from positive matches. The equations of this loss metric are (2), (3), (4), (5) and (6). Localization loss is the loss during the prediction of the target classification, which is calculated using (7) and (8). These two loss functions are later combined in a general one by (9) to communicate the loss of the DNN in a single parameter.

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in (cx,cy,w,h)} x_{ij}^{k} smooth_{L1}(l_i^m - \hat{g}_j^m) \qquad (2)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^{w} \qquad (3)$$

$$\hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^{h} \qquad (4)$$

$$\hat{g}_j^{w} = log(\frac{g_j^{w}}{d_i^{w}}) \qquad (5)$$

$$\hat{g}_j^{h} = log(\frac{g_j^{h}}{d_i^{h}}) \qquad (6)$$

$$L_{conf}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^{p} log(\hat{c}_i^p - \sum_{i \in Neg} log(\hat{c}_i^0) \qquad (7)$$

$$\hat{c}_i^p = \frac{exp(c_i^p)}{\sum_p exp(c_i^p)} \qquad (8)$$

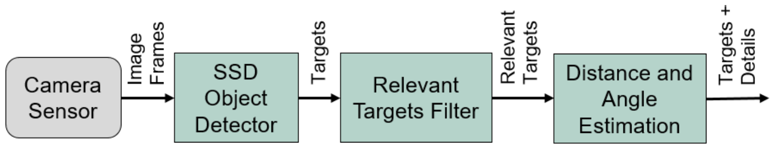


FIGURE 4. Camera data processing pipeline.

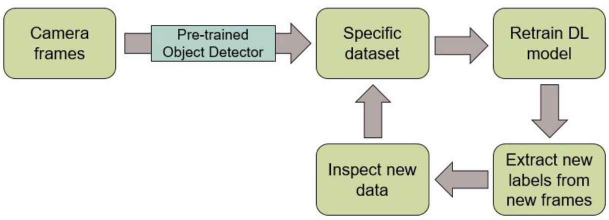


FIGURE 5. Dataset creation.

$$L(x, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \tag{9}$$

where N indicates the number of matches default boxes, l represents the predicted boxes, g the ground truth boxes, x the coordinates of the bounding boxes and c the classes confidences. These parameters also include the offset for the center points (cx, cy), the width of the box (w) and its height (h).

The object detector was built based on public pre-trained models available in multiple DL libraries such as MXNet [25] or TensorFlow Lite [26]. These DL framework were designed by different companies but they share the same goal, to reduce the resource requirements during the training and inference phases. In this paper we will not aim to train the DL model at the network edge. Nevertheless, these frameworks increase the efficiency of the inference process and reduce the model size.

The pre-trained DL model was used to extract a first iteration of the labels from a reduced number of frames from the camera data. These labels needed to be filtered in order to remove not relevant classes as well as adding new classes. After this step, the new labels are used to train a new DL model for object detection using transfer learning. This technique was used due to the reduced dimension of the new dataset. This new model will be used to generate labels from more frames, increasing the number of samples in our specific dataset as shown in Figure 5. This iterative process can lead to exponential errors if the labels are not correct. Therefore, an inspection of the new data will be executed after the first re-training phases, which are the most critical. This ensures the correct application of this approach. The iterative process will finish once the dimensions of the dataset are large enough to ensure an accuracy of the model over 90% in the data of our experiment.

After training our object detector, labels from the camera data can be generated automatically. These labels may not be relevant for the sensor fusion, due to the fact that the camera may see objects which are not in the range of the radar or they are not relevant for the sensor fusion. As a result, these labels need to be preprocessed to remove non relevant targets before the data fusion stage.

Once the localization of the targets is extracted, it is possible to estimate the angle of arrival of those objects based on their coordinates. The resolution of the camera images plays a key role in this process since it is directly proportional to the precision of the AoA estimation in the camera data. This process is based on assuming an AoA of 0° for targets located in the center of the image and it increases as the targets moves to the right side of the camera frame, as the AoA extracted from the radar data.

In order to ensure the correct measurement of this parameter, these values have been calibrated with the AoA from the radar data. At the same time, the distance of the targets to the camera sensor can be extracted after a calibration based on the specific scenario. As a result of this, the X and Y coordinates of the targets in the camera data can be transformed into distance and AoA measurements. These features will be used for the data fusion process further explained in the next subsection.

C. DATA FUSION

The previous preprocessing techniques provide relevant features from the raw camera and radar data. These features are shown in Table 1.

Before fusing the data, a correct synchronization of the data from both sensors is crucial. To enable this, timestamps have being added to the data during the recording phase. However, the frame rate of both sensors may not be the same. Therefore, the sensor with the highest latency will be used as the standard latency of the system during the synchronization. A possible delay in the timestamps between the sensors has been taking into account during the synchronization step. Therefore, the data will be matched with the nearest data in the time domain.

The labeling process is triggered when a target is detected in the camera data. Therefore targets that are not detected by this sensor, such as highly covered targets or partially visible targets may not be labeled. At this point, the targets localized in the camera are compared with the clusters from the RDM image based on the distance of the targets respect the recording system. The distance between the camera and radar sensors of the system is reduced enough to be able to be ignored in comparison with the distance of the studied targets.

TABLE 1. Features extracted from each sensor data.

	Features			
Radar sensor	Cluster	AoA in radar	Speed	Distance
Camera sensor	Classification	AoA in camera	Localization	Target groups

One more limitation of the framework is the difference in the Field of View (FoV) of both sensors. Since our approach is based on comparing the targets found in each sensor's FoV, if the overlapping of the FoV of the studied sensors is not total, there may be targets that are not found by all the sensors. In order to overcome this, only the intersection FoV of these sensors has been studied for the label creation. Another implemented approach to reduce the impact of this limitation is the use of labels from previous frames. By knowing where a target was in a previous frame as well as its speed (extracted from the RDM), it is possible to estimate where it will be located in the current frame. The estimation of the location of the known targets is used to locate relevant areas in the RDM image as well as clusters what may have not being detected initially or when the reflection power of the radar signal was not powerful enough in that frame.

At the same time, to distinguish targets at the same distance, the measured AoA from both sensors is used as auxiliary parameter to ensure the correct labeling. Because of the difference in the resolution of both sensors, an error range of 5° has been included in the algorithm. As a result, the AoA from both sensors does not have to be exactly the same. This approach solves multiple problems due to angle offset between the sensors as well as noise in the data.

There are multiple algorithms to extract the AoA from the radar data in the literature. Among all the methods, the beamforming method [27] was chosen for the experiment. In this technique, the angle is calculated by coherently summing the signals from different receiving channels.

Since this tool should be understood as a first approach for the dataset labeling process, it also generates a text file with relevant information after its execution. This file contains information such as the path for the data and the number of the frames that need to be reviewed. The revision may be necessary because of targets in the camera data that did not match with any cluster of the RDM. This will only be reported when these targets are located in an area where the radar should be able to detect them. Similarly, it will also report cases where there are unmatched clusters in the RDM. As a result of this, the reviewing process can be executed efficiently rather than having to study all the labeled frames.

In order to evaluate the performance of the proposed framework, an experiment has been executed and explained in the next section.

IV. EXPERIMENT

In this section, we evaluate the designed framework on a custom proprietary dataset created by Infineon specifically for this application. Our goal is to compare the accuracy

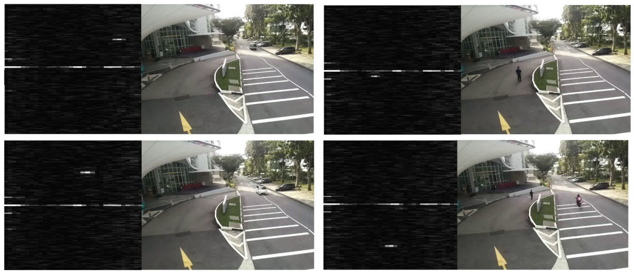


FIGURE 6. Samples of pairs of images used to label the RDM images.

of the labeling process and its latency with a traditional approach of labeling the data manually as well as some state-of-the-art techniques. The techniques used for the comparison are described in [1], [2], [11], [28]. These baseline techniques were selected due to the state-of-the-art results obtained in similar fields as the one research in our paper. Next, we describe the dataset and how it was built.

A. DATASET

The previously explained framework was tested to label data from radar sensors in a vehicle context where all the studied frames include relevant targets for the application. This main dataset was generated with data from two different locations: Singapore Polytechnic Campus, shown in Figure 6, and Infineon Singapore Campus, shown in Figure 7. The height of the sensors in these locations was different to fit the location as well as gather data from different conditions to create a general dataset. The height of the sensors was 5 and 2.3 meters for the chosen locations respectively.

The relevant categories that have been used for the classification of the labels are: pedestrian, car, truck, bicycle, motorbike and personal mobility device (PMD). The last category was not included in any public-pretrained object detector leading to the test of the addition of new categories in the pipeline. The distribution of the classes has been studied to ensure there are no imbalanced classes during the training phase of the DL model.

The Figure 6 shows some examples of the pair of images from both sensors included in the mentioned dataset. The initial labels for our dataset have been created manually including target classification and 2D bounding boxes.

An additional dataset was generated using different radar and camera sensors as well as a different scenario to test the tool developed with the previous dataset. This reduced dataset contains 50 frames with 2 target classes (vehicle and obstacle). In this case, the sensors were integrated in a reduced scale vehicle where the height of the sensors was

30 centimeters to the floor. An example of camera and radar RDM input data from the second dataset is shown in Figure 8.

B. HARDWARE ARCHITECTURE

Both sensors researched in our paper, in the main dataset, for the Sensor Fusion were controlled by a Rasberry Pi 3 device. This device was used to ensure the synchronization of the data during the data gathering.

As previously explained, the radar sensor used in the main dataset is a Multiple-input-multiple-output (MIMO) frequency modulated continuous wave (FMCW) radar with four channels developed by Infineon Technologies AG. This radar sensor can detect targets until a distance of 40 meters with a resolution of 1.25 meters because of its configuration, what make suitable for this application. Depending on the manufacturer, the initial parameters of the radar sensor may variate from the one used in this experiment. However, RDM images can be generated with other configurations resulting in the RDM format studied in this paper. In our case, the configuration of the radar sensor used in the main dataset is 24 GHz for the center frequency, 200 MHz for the bandwidth, 64 samples per chirp and 256 chirps per frame.

The camera sensor used in the main dataset is an optical camera for Raspberry Pi model IMX219PQ. This camera sensor can record data at a maximum frame rate of 60 frame/second. However, the frame rate used in this project is 6 frames/sec in order to reduce the memory consumption of the data. Multiple recording sessions in two different scenarios were executed to collect enough data for the training and evaluation of the algorithms.

C. DEEP LEARNING MODEL

An iterative process has been executed in order to train an object detector for the camera data as explained in Subsection 3.2. This approach has been followed due to the fact that new classes, which were not present in previous object detectors, have been included for this experiment. The pre-trained model used was a SSD structure based on ResNet50 backbone.

This pre-trained model was used to extract 774 labels from 100 camera images. These labels were filtered to correct labels corresponding to new classes, in this case PMD, as well as removing wrong labels. These labels were used to re-train the model in order to fit our specific application achieving an accuracy of 46.72% in the first iteration (measured as the correct predictions over the wrong predictions and false negative results). The iterative process was executed 10 times adding 100 new camera pictures in each iteration. The information about the last iteration of the re-training phase of the model is shown in Table 2.

The DL library used for the training of this model was MXNet because of its focus on low-resource devices. This leads to optimize the DNN during the design phase, leading to a faster inference process.

TABLE 2. Results and specifications of the object detector.

Size of the model	TP	FP	FN	Accuracy
132.18 KB	961	27	12	92.1%

FIGURE 7. (a) Image extracted from the Object detector using the camera data. (b) Result of the sensor fusion approach proposed in this paper.

D. EVALUATION

Once the labels for the camera data are generated with this new DL model, the distance of the targets has been estimated based on some camera calibrations. These calibrations consists on the comparison of manually labeled targets in the RDM as well as the camera frames to determine the relationship between the coordinates in the camera image and the real distance. At the same time, the synchronization of the data was ensure by the comparison of the timestamps of the data.

The Intersection over Union (IOU) technique has been used to compare the predicted bounding box labels of the proposed framework and the ground truth labels. IOU is an evaluation metric commonly used to measure the accuracy of object detector DNNs. This technique can be applied to any system that predicts bounding boxes in scenarios where the ground truth is known. An IOU result above 0.5 is normally considered as a good prediction. The algorithm itself is explained in (10) where A means the ground truth bounding box and B the predicted bounding box.

$$IOU = \frac{A \cap B}{A \cup B} \tag{10}$$

Using the features previously explained as well as the rule system explained in Subsection 3.3, the data from both sensors was fused to generate labels for the RDM data, as shown in Figure 7. The accuracy results obtained using the proposed framework are shown in Table 3. The mean average precision (mAP) is also shown in this table. The global system works as an object detector for the RDM images, locating relevant targets as well as classifying them. Therefore, in Table 3 only the true positive, false positive and false negative are studied due to the fact that there is no true negative results in this approach.

CPU and GPU time gain cannot be accurately compared with objective measures since a CPU and a GPU have a very different hardware architecture. However, these times can be used as an orientation of the time the tool requires for the labeling process. The CPU and GPU used are the E5-2643 v3 and a NVIDIA Tesla P4 GPU respectively. The

TABLE 3. Results of the automatic label creation process.

Average speed of process	TP	FP	FN	mAP
3 frames/sec (GPU)	2474	541	2079	82.056%
0.667 frames/sec (CPU)	2474	541	2079	82.056%

TABLE 4. Results of the automatic label creation process in each of the scenarios.

Location	mAP
Singapore Polytechnic Campus	85.260%
Infineon Singapore Campus	80.730%

FIGURE 8. Camera and radar range Doppler map image from the second dataset.

speed of the tool in these platforms, including all the required preprocessing of the data before the labeling process, has been shown in Table 3. These times can also be used to test the possibility of using small Edge AI devices equipped with GPU capability to achieve a good performance in time complexity being compared even with a desktop CPU.

These results can be further analyzed by studying the accuracy achieved by the proposed tool in each of the scenarios of the dataset, as shown in Table 4. This table shows how the accuracy in both scenarios is similar but the results in the Singapore Polytechnic Campus are higher. This might be due to the fact this scenario has less objects near the sensors that could difficult the target location as well as provide a more general view of the scenario.

The second dataset was used to test the developed approach. Consequently, the tool was also evaluated in the 50 frames of the second dataset which was recorded using a different radar sensor from the main dataset. This second dataset's radar is the Infineon's BGT60TR13C 60 GHz radar sensor [29]. The configuration of this sensor was 60.7 GHz for the center frequency, 1 GHz for the studied bandwidth, 128 samples per chirp and 64 chirps per frame. The camera sensor used for the data gathering of this dataset was a 5 MP Raspberry Pi camera, with a viewing angle of 160°. An example of the data contained in this dataset is shown in Figure 8.

The final accuracy achieved in this dataset was 87.61%. This result should be understood as a proof of the suitability of the proposed framework for multiple sensors and scenarios.

In order to compare the time reduction achieved by using this automatic labeling tool, the same dataset has been labeled manually and using the proposed tool. The required time to label 400 frames manually was 5 hours. On the other hand, when using the proposed autolabeling framework, the required time for the same task was reduced to 6 minutes (using a GPU platform). Therefore, the time reduction achieve in this dataset was 96.76% respect to the manual labeling process. When using a CPU platform, the time required to use the tool increased, leading to an inferior time reduction of 85.01% respect to the manual labeling process.

TABLE 5. Comparison of results achieves with other techniques.

Technique	initial mAP	Average speed
Our approach	82.056%	3 frames/sec
K. Patel et al. [2]	65.30%	2 frames/sec
F. Nobis et al. [1]	57.50%	–
T.Y. Lim et al. [16]	73.5%	40 frames/sec
T. Winterling et al. [11]	94.93%	–

Other emerging techniques for object detection based on radar and camera, such as the technique proposed in [2], were able to achieve higher accuracy than our technique. However, these results are based on the study of multiple frames for the target tracking to enable high accuracy results. If the results are compared only when the system study a single frame, the initial accuracy of [2] is reduced considerably to the point where the accuracy of our technique outperforms it. The time required to process the data has also been improved in our technique in comparison with the K. Patel *et al.* technique [2].

As shown in Table 5, multiple authors do not include the speed of their proposed algorithms in their papers. Therefore, the latency of our system can only be compared with the technique proposed in [2] and [16].

The results obtained in [11], as well as the results from [2] (if we include the tracking system to study multiple frames), achieved a higher accuracy than our proposed framework. However, [11] is able to achieve these results through an iterative approach where human interaction is still required to manually label part of the dataset in each iteration. At the same time, the technique was not tested with multi-class data, in contrast with our technique which is able to detect multiple targets in the same frame. Due to the fact that our goal is to create a fully automatic process, the human interaction in our technique is minimal but this is a trade off with the final accuracy achieved.

On the other hand, in [2], the high performance results were obtained after applying a tracking technique to correct wrong classification as well as unclassified objects. This tracking system also increases the complexity as well as the memory consumption of this technique. In a first stage, the average accuracy results obtained are 62.95%, where our technique outperforms achieving an 82.06% accuracy. It is important to remark the techniques compared with our approach were tested under a different dataset since NuScenes dataset does not provide range Doppler maps, which are required for our framework.

The proposed pipeline by F. Nobis in [1] is based on extracting 2D points from the radar data before the fusing process, what enable its direct compatibility with other sensors such as LiDAR sensors. This generalization of the input data format as well as their Early Fusion approach may be some of the reasons of its low accuracy results (57.50%) in comparison with out tool (82.056%) and the rest of algorithms compared. However, it is important to mention this approach may not provide high accuracy as other techniques but enables its implementation in a wider sensor scenario.

The same conclusion can be extracted from the proposed algorithm by Lim *et al.* [16]. Since this author also designed an Early Fusion technique based on DNNs, the limitations of their technique is the same as [1]. However, this author achieves a higher accuracy than [1] because of the different data format enabling a pyramid network structure. Nevertheless, this approach still achieves lower accuracy than the rest of the compared algorithms due to the lack of control over the preprocessing of the data and the evaluation of these preprocessing techniques. On the other hand, due to the fact that this algorithm does not execute a previous deep preprocessing, the system's speed is higher than other techniques, achieving 40 frames/sec. Therefore it is possible to see this as a trade-off between accuracy and frame rate. Nevertheless, we focus on achieving high accuracy results (8.556% higher than the T.Y Lim *et al.* technique [16]) since accurate labels are required, otherwise the usefulness of the framework would be decreased.

It is also important to remark all the previously commented approaches predict bounding boxes for targets in camera data. Therefore the radar data is used as support/reinforcement data. On our framework, the goal is to generate bounding boxes for the RDM images using the camera data to extract relevant features for this task. As a consequence of this, the previous comparisons are used as an orientation of the accuracy in comparison with the state of the art.

V. CONCLUSION

An efficient Sensor Fusion framework to automatically generate labels for range Doppler maps has been described in this paper. An experiment where this framework has been evaluated has been explained as an use-case of this framework for the industry. The tool provided high accuracy results while maintaining low-resource requirement and low latency in this experiment.

The proposed technique is based on multiple state-of-the-art sensor fusion algorithms, extracting the advantages from each of them to further improve the system. Difficulties and solutions to process the required data in our algorithm have been discussed in this paper. We show that the fusion of radar and camera data does not require complex structures to achieve high accuracy results while maintaining low latency. This lends justification to a variety of new sensor fusion algorithms where the algorithms are optimized for radar and camera sensor.

The proposed pipeline approach in our paper is flexible enough to be applied with other sensors and environment conditions. As an example, LiDAR sensors could be used instead of camera sensors to label the data to train the stand-alone radar-based target detection system. This could overcome traditional camera/vision sensor approach regarding adverse environmental conditions (i.e. lighting, reflections, etc.). In future works, we will attempt to fuse data coming from different heterogeneous sensors, such as LiDAR, radar and camera, to overcome the sensor limitations in environment data gathering.

All the experiments to test the proposed labeling tool were executed in multiple computer platforms as previously described in Section 4. However, the final goal of this tool is to be implemented at the network edge. For this reason, the AI model integrated in the proposed pipeline was developed taking into account the restrictions of memory available in Edge devices. Therefore, it would be possible to move its execution to the network edge where it could execute the data gathering and labeling simultaneously if the Edge device where it is implemented has enough resources.

ACKNOWLEDGMENT

The company Eesy Innovation collaborated during the experimental and test phase of this project by testing the proposed tool in this article in different scenarios to check the accuracy of the system in multiple environments.

REFERENCES

[1] F. Nobis, M. Geisslinger, M. Weber, J. Betz, and M. Lienkamp, "A deep learning-based radar and camera sensor fusion architecture for object detection," in *Proc. Sensor Data Fusion, Trends, Solutions, Appl. (SDF)*, Bonn, Germany, Oct. 2019, pp. 1–7.

[2] K. Patel, K. Rambach, T. Visentin, D. Rusev, M. Pfeiffer, and B. Yang, "Deep learning-based object classification on automotive radar spectra," in *Proc. IEEE Radar Conf. (RadarConf)*, Boston, MA, USA, Apr. 2019, pp. 1–6.

[3] H. Han, J. Kim, J. Park, Y. Lee, H. Jo, Y. Park, E. T. Matson, and S. Park, "Object classification on raw radar data using convolutional neural networks," in *Proc. IEEE Sensors Appl. Symp. (SAS)*, Sophia Antipolis, France, Mar. 2019, pp. 1–6.

[4] J. Lombacher, M. Hahn, J. Dickmann, and C. Wohler, "Object classification in radar using ensemble methods," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Nagoya, Japan, Mar. 2017, pp. 87–90.

[5] N. Del-Rey-Maestre, M.-P. Jarabo-Amores, D. Mata-Moya, J.-L. Barcena-Humanes, and P. G. D. Hoyo, "Machine learning techniques for coherent CFAR detection based on statistical modeling of UHF passive ground clutter," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 104–118, Feb. 2018.

[6] S. Zhu, J. Xu, H. Guo, Q. Liu, S. Wu, and H. Wang, "Indoor human activity recognition based on ambient radar with signal processing and machine learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6.

[7] L. Li, Y. Lv, and F.-Y. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA J. Autom. Sinica*, vol. 3, no. 3, pp. 247–254, Jul. 2016.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013, *arXiv:1312.5602*. [Online]. Available: http://arxiv.org/abs/1312.5602

[9] D. Zhukov, J.-B. Alayrac, R. G. Cinbis, D. Fouhey, I. Laptev, and J. Sivic, "Cross-task weakly supervised learning from instructional videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3537–3545.

[10] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, Mar. 2019.

[11] T. Winterling, J. Lombacher, M. Hahn, J. Dickmann, and C. Wöhler, "Optimizing labelling on radar-based grid maps using active learning," in *Proc. 18th Int. Radar Symp. (IRS)*, Jun. 2017, pp. 1–6.

[12] J. Tang and P. H. Lewis, "A study of quality issues for image auto-annotation with the Corel dataset," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 3, pp. 384–389, Mar. 2007.

[13] M. D. Cicco, C. Potena, G. Grisetti, and A. Pretto, "Automatic model based dataset generation for fast and accurate crop and weeds detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2017, pp. 5188–5195.

[14] M. Suchi, T. Patten, D. Fischinger, and M. Vincze, "EasyLabel: A semi-automatic pixel-wise object annotation tool for creating robotic RGB-D datasets," in *Proc. Int. Conf. Robot. Automat.*, May 2019, pp. 6678–6684.

[15] G. Meseguer-Brocal, A. Cohen-Hadria, and G. Peeters, "DALI: A large dataset of synchronized audio, lyrics and notes, automatically created using teacher-student machine learning paradigm," Jun. 2019, *arXiv:1906.10606*. [Online]. Available: https://arxiv.org/abs/1906.10606

[16] T.-Y. Lim, A. Ansari, B. Major, D. Fontijne, M. Hamilton, R. Gowaikar, and S. Subramanian, "Radar and camera early fusion for vehicle detection in advanced driver assistance systems," in *Proc. Mach. Learn. Auton. Driving Workshop 33rd Conf. Neural Inf. Process. Syst.*, 2019, pp. 1–11.

[17] Z. Ji and D. V. Prokhorove, "Radar-vision fusion for object classification," in *Proc. 11th Int. Conf. Inf. Fusion*, 2008, pp. 1–7.

[18] J. Kocic, N. Jovicic, and V. Drndarevic, "Sensors and sensor fusion in autonomous vehicles," in *Proc. Belgrade, Telecommun. Soc. Academic Mind (TELFOR)*, Nov. 2018, pp. 420–425.

[19] X. Zhang, M. Zhou, P. Qiu, Y. Huang, and J. Li, "Radar and vision fusion for the real-time obstacle detection and identification," *Ind. Robot, Int. J. Robot. Res. Appl.*, vol. 46, no. 3, pp. 391–395, May 2019.

[20] P. Duygulu, K. Barnard, J. F. G. de Freitas, and D. A. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," in *Proc. Eur. Conf. Comput. Vis.*, vol. 2353, Apr. 2002, pp. 97–112.

[21] H. Rohling, "Radar CFAR thresholding in clutter and multiple target situations," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-19, no. 4, pp. 608–621, Jul. 1983.

[22] F. C. Robey, D. R. Fuhrmann, E. J. Kelly, and R. Nitzberg, "A CFAR adaptive matched filter detector," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, no. 1, pp. 208–216, 1992.

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, in Lecture Notes in Computer Science, 2016, pp. 21–37.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 91–99.

[25] *MXNet Deep Learning Framework*, Apache, Forest Hill, MD, USA. Accessed: Apr. 20, 2020.

[26] *Tensorflow Lite Deep Learning Framework*, Google, Mountain View, CA, USA. Accessed: Apr. 20, 2020.

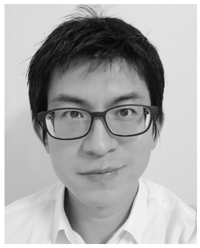
[27] S. Chandran, "Smart antennas for wireless communications (with MATLAB) (gross, F.; 2005) [reviews and abstracts]," *IEEE Antennas Propag. Mag.*, vol. 51, no. 3, p. 134, Jun. 2009.

[28] K.-E. Kim, C.-J. Lee, D.-S. Pae, and M.-T. Lim, "Sensor fusion for vehicle tracking with camera and radar sensor," in *Proc. 17th Int. Conf. Control, Autom. Syst. (ICCAS)*, Jeju, South Korea, Oct. 2017, pp. 1075–1077.

[29] T. Stadelmayer, A. Santra, R. Weigel, and F. Lurz, "Parametric convolutional neural network for radar-based human activity classification using raw ADC data," *IEEE Sensors J.*, 2020. [Online]. Available: https://www.techrxiv.org/articles/preprint/Parametric_Convolutional_Neural_Network_for_Radar-based_Human_Activity_Classification_Using_Raw_ADC_Data/12896108/2

JAVIER MENDEZ received the B.Sc. degree in electronics engineering from the University of Granada, in 2018, and the M.Sc. degree in electronics, robotics and automatics engineering from the University of Sevilla, Spain, in 2019. He is currently pursuing the Ph.D. degree with Infineon in collaboration with the University of Granada and worked on artificial intelligence at the network edge as well as sensor fusion algorithms. His current research interests include edge computing, deep learning, and sensor fusion. He received the Head of the Graduating Class Award by the University of Sevilla.

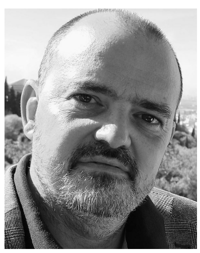
STEPHAN SCHOENFELDT received the M.Sc. degree in communication engineering from the University of Applied Science in Frankfurt, Germany, in 1999. He joined Infineon Technologies AG, where he filled different roles covering product and application engineering as well as digital design for application specific CMOS devices in the field of data storage and medical SOCs. In 2007, he took the role of a Concept Engineer working on algorithms and verification methods for power conversion systems. Since 2017, he has been developing algorithms for radar sensors in the industrial and consumer application domain with the focus on machine learning approaches.



XINYI TANG (Member, IEEE) received the B.S. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2006, and the Ph.D. degree from the National University of Singapore, Singapore, in 2011. He worked in universities and companies for various projects related to phased array communication systems and MIMO radar systems. Since 2020, he has been with the Signal Processing, RF and Optical (SRO) Department, Institute for Infocomm Research, Agency for Science, Technology and Research, as a Scientist. His current research interests include software defined radio applications, mmwave RF sensing, and communication systems. He was a recipient of the Best Student Poster Paper Award of the 2007 IEEE International Conference on Electron Device and Solid-State Circuits.

JAKOB VALTL received the B.Sc. degree in engineering science and the M.Sc. degree in electrical engineering and information technology from the Technical University of Munich, in 2016 and 2020, respectively. He is currently pursuing the Ph.D. degree with Infineon in collaboration with the Technical University of Braunschweig and worked on artificial intelligence on edge devices in particular on adversarial attacks on radar data and secure algorithms.

M. P. CUELLAR graduated in computer engineering, in 2003. He received the Ph.D. degree in time series prediction, parameter identification, and neural networks, in 2006. He is currently a full-time Teacher with the Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain. His main interests include neural and social networks, evolutionary optimization, and fuzzy systems. He has also worked in multivariate image analysis and real-time control tasks.



DIEGO P. MORALES received the B.Sc., M.Eng., and Ph.D. degrees in electronics engineering from the University of Granada, in 2001 and 2011, respectively. Since 2001, he has been an Associate Professor with the Department of Computer Architecture and Electronics, University of Almeria, before joining the Department of Electronics and Computer Technology, University of Granada, in 2006, where he currently serves as a Tenured Professor. He is the Co-Founder of the Biochemistry and Electronics as Sensing Technologies (BEST) Research Group, University of Granada. He has coauthored more than 80 scientific contributions. His current research interests include low-power energy conversion, energy harvesting for wearable sensing systems, and new materials for electronics and sensors.