Association of Camera and Radar Detections Using Neural Networks

Konstantinos Fatseas
Computer Architectures for Embedded Systems
University of Twente
Enschede, The Netherlands
k.fatseas@utwente.nl

Marco J.G. Bekooij
Embedded Software and Signal Processing
NXP Semiconductors
Eindhoven, The Netherlands
marco.bekooij@nxp.com

2023 IEEE Radar Conference (RadarConf23) | 978-1-6654-3669-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/RADARCONF2351548.2023.10149729

***Abstract*—Automotive radar and camera fusion relies on linear point transformations from one sensor's coordinate system to the other. However, these transformations cannot handle non-linear dynamics and are susceptible to sensor noise. Furthermore, they operate on a point-to-point basis, so it is impossible to capture all the characteristics of an object.**

This paper introduces a method that performs detection-to-detection association by projecting heterogeneous object features from the two sensors into a common high-dimensional space. We associate 2D bounding boxes and radar detections based on the Euclidean distance between their projections. Our method utilizes deep neural networks to transform feature vectors instead of single points. Therefore, we can leverage real-world data to learn non-linear dynamics and utilize several features to provide a better description for each object.

We evaluate our association method against a traditional rule-based method, showing that it improves the accuracy of the association algorithm and it is more robust in complex scenarios with multiple objects.

***Index Terms*—camera, radar, sensor fusion**

I. Introduction

Automotive radars and cameras are cost-effective and production-ready sensors. They complement each other and provide a great deal of information for objects within their sensing range. On the one hand, vision sensors offer accurate semantic data and have a high angular resolution but cannot directly measure an object's velocity and radial distance. On the other hand, radars are robust and accurate when measuring the distance to an object and its radial velocity. However, they cannot provide reliable semantic information and have a lower angular resolution. Consequently, matching detections from the two sensors improves our perception by combining sensing modalities.

Automotive sensor fusion algorithms merge data from different sensors in an early or later stage. What differentiates those methods is the amount of preprocessing on the raw output of each sensor. For example, early fusion algorithms merge raw or unfiltered data streams to improve object detection. Usually, a deep neural network (DNN) is fed with radar data and RGB images and performs object detection in the radar, or the image coordinate system [1], [2]. On the other hand, late fusion requires preprocessing and combining the data from the two modalities on an object level [3], [4]. Our method falls into the second category as we match objects from the two modalities after object detection.

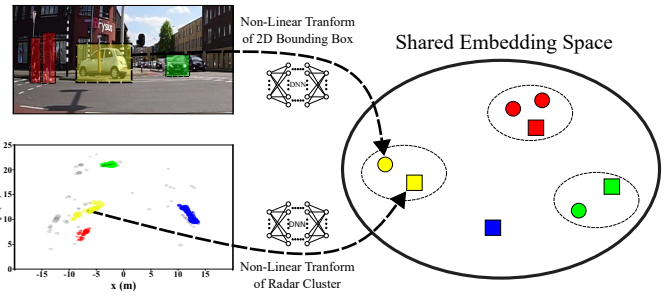

Fig. 1. Overview of our detection association method. We train neural models to project features of 2D bounding boxes and radar reflection clusters in a shared embedding space. The camera and radar detections are then associated based on the Euclidean distance within the learned space. We illustrate a bounding box's projection as a circle and the projection of a radar cluster as a square. The color of each shape indicates the object it represents.

Existing methods allow us to perform object detection on each sensor's output reliably. However, matching the detections from different modalities remains an open problem because of heterogeneity in data. Traditional methods use a distance metric to perform detection-to-detection association. For example, following sensor calibration [5], we can transform a point in one sensor's coordinate system to the coordinate system of the other sensor and associate the close detections from the different modalities based on their Euclidean distance. However, these methods are vulnerable to noise and cannot capture non-linear dynamics as they rely on linear transformations. Furthermore, describing a detected object with a single point results in losing information that could otherwise improve the association. For example, the extent of an object, its class, its velocity signature, and the strength of its reflection is information that can better characterize the object. Nevertheless, it remains unused in a point-to-point transformation.

In this work, we propose a method that combines traditional rule-based algorithms with the flexibility of neural networks. Our method for data association relies on deep metric learning. In doing so, we retain the metric-based association of the detected objects across the two modalities. The metric is the distance between detections within the so-called embedding space, which the neural networks learn from real-world data. In other words, we use neural models to transform the features

Authorized licensed use limited to: UNIVERSITY OF TWENTE.. Downloaded on March 13,2024 at 14:56:05 UTC from IEEE Xplore. Restrictions apply.

of objects detected with the camera and the radar into a shared space where we can directly compare them.

Our learning-based approach uses neural networks to model each sensor and its noise and adapt to occlusions and other corner cases. More importantly, the neural models we train to project detections to the high-dimensional embedding space consider multiple features. Therefore, we provide a better description for each object and we transform them as a whole rather than a single point. As a result, the learned association metric is robust and improves the overall accuracy of the association algorithm.

Cross-modal matching based on deep metric learning has been used to associate text and images [6]. Our work follows the same principle. It utilizes two separate neural networks that transform the heterogeneous data from the camera and radar sensing modalities to a shared embedding space. Therefore, each modality has a dedicated neural model that projects its detections to the shared space. To train them jointly, we use the triplet loss function [7], which forces the neural models to generate a similar output (embedding vector) when the detection originates from the same object in the real world.

As seen in Figure 1, we replace the sensors' coordinate system with a learned space where the corresponding neural model projects each detection from the radar or the camera without losing much information as in the point-to-point transformation. The Euclidean distance between their embedding vector determines whether two detections correspond to the same object. After calculating the distance between all object embeddings and computing a similarity matrix, we solve the assignment problem using the nearest neighbor algorithm.

Our main contributions are:

- We show the feasibility of cross-modal deep metric learning for radar and camera detection matching.
- We show that we can use the distance within the learned embedding space for associating objects from the two modalities. With the learned metric, we improve the performance of the assignment algorithm compared to the traditional rule-based method.

The remainder of this paper is structured as follows. In Section II, we discuss other radar and camera sensor-fusion methods. In Section III, we explain the preprocessing and object detection stages required before object matching. We also present the training procedure and operation of our method during inference. In Section IV, we provide details on the dataset we used for training and testing. Furthermore, we evaluate our object matching method by comparing it with a baseline algorithm. Finally, we provide our conclusion in Section V

II. Related Work

Deep neural networks are very flexible regarding input data shape and type. Therefore, a common way of fusing radar and camera data in the early stages is by feeding a convolutional neural network with radar and camera images. The neural network internally combines the two data streams to enhance object detection. In [1], Lim et al. propose a deep neural architecture that processes unfiltered range-angle and camera images. After several initial convolutional layers, they transform the intermediate representations into the so-called bird's eye view. Following more convolutional layers, the spatially aligned feature maps from the two streams are stacked together and fed to a detection head. A single-shot detector performs multi-object detection on the fused feature maps and predicts 2D bounding boxes in the bird's eye view plane. This plane is essentially a rasterized version of the real-world coordinate system.

Nobis et al. present another neural-network-based early fusion architecture in [2]. In their work, a neural network processes the camera image and a synthetic image containing radar detections. What differentiates this work from [1] is that instead of projecting the camera image into the radar's coordinate system, the authors project radar detections into the camera pixel coordinate system. Consequently, the object detection head predicts bounding boxes within the camera image and uses radar data to enhance its detection performance in adverse weather conditions.

The studies above utilize neural networks to process radar and camera data concurrently without heavy preprocessing. The main goal of such methods is to improve the system's overall perception. In contrast, our work is related to late fusion techniques, where we associate detected objects from the two modalities.

For example, Aziz et al. introduce a detection to detection fusion algorithm in [3]. However, in their work, neural models are only used for object detection and classification, not fusion. They perform detection to detection association by projecting radar detections into the camera plane after estimating the homography transform using several reflectors. In [4], the authors perform sensor fusion by calculating the angle and distance for each bounding box detected in the visual modality and comparing it with the readings of the radar sensor.

In a work closer to ours, Dong et al. [8] utilize deep metric learning for camera-radar object association by projecting 2D bounding boxes and radar detections into the camera image. A deep neural network then processes the resulting pseudo-image and outputs several vectors whose number equals the total number of objects detected in both modalities. To perform the association, they use the distance between the output vectors.

Contrary to [8], our method does not rely on a single convolutional DNN but a separate lightweight neural model for each modality. Furthermore, our association method does not require manual sensor calibration in contrast to the previously mentioned papers. Instead, we expect the two neural networks to bridge the modality gap by learning a transformation of each modality to a shared embedding space. Additionally, the flexibility of neural models allows us to consider features that remain unused when the object is described by a single or even several points. As a result, our method offers the simplicity of late fusion techniques but is also flexible and capable of using lower-level features.

Authorized licensed use limited to: UNIVERSITY OF TWENTE.. Downloaded on March 13,2024 at 14:56:05 UTC from IEEE Xplore. Restrictions apply.

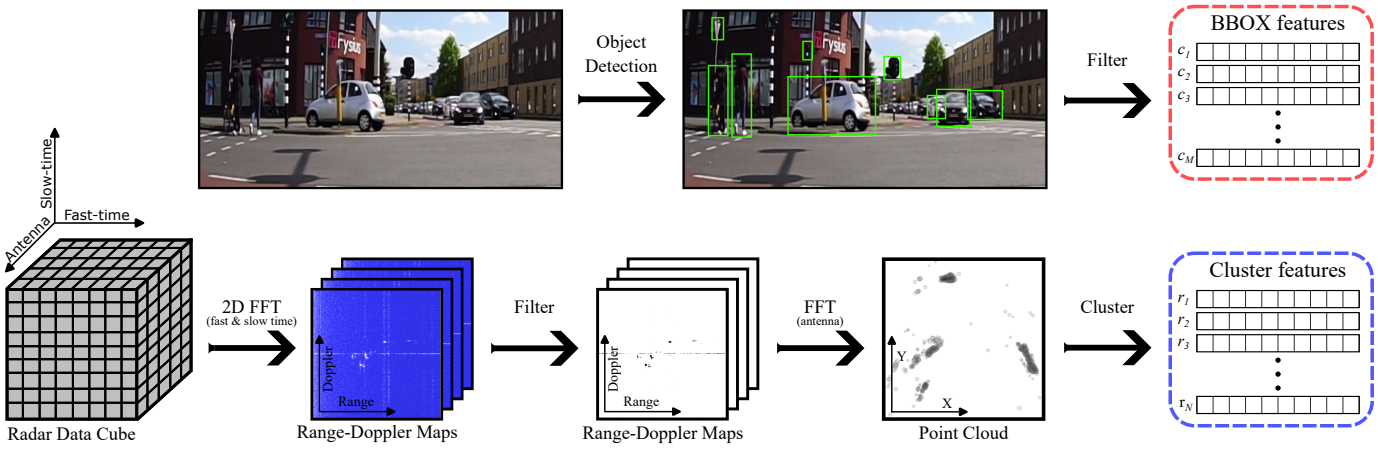

Fig. 2. Preprocessing of the radar and camera data.

III. Methodology

In this section, we describe our detection matching method. First, we discuss the preprocessing steps to acquire a list of detected objects from each modality. Then, we formulate the matching problem and show how we jointly train two neural models to project the detections into a shared space. Finally, we describe the matching process based on the Euclidean distance between the embedding vectors of all objects.

A. Data Preprocessing

The proposed method fuses camera and radar data at an object level. Therefore, it requires a list of detected objects from both sensors. We illustrate the preprocessing stages for both modalities in Figure 2. In the upper branch, we show the detection pipeline for the visual modality. We utilize an object detector of the YOLO family [9] to extract a list of relevant objects, such as pedestrians, cyclists, and vehicles. A 2D bounding box, defined by its pixel coordinates, indicates each object's location within the image. Additionally, for each detection, we get its class and a confidence score which help filter out irrelevant objects. In the right column of Table I, we list the features that describe each detection in the visual modality and we feed to the corresponding neural model in the next stage of our method.

TABLE I
List of features used to describe radar and camera detections.

	Radar Cluster Features	*Camera BBox Features*
1	min range	left pixel coordinate
2	max range	right pixel coordinate
3	mean range	area
4	min radial velocity	width
5	max radial velocity	height
6	mean radial velocity	height^{-1}
7	min DoA	class
8	max DoA	
9	mean DoA	
10	min amplitude	
11	max amplitude	
12	mean amplitude	
13	size (number of points)	

On the radar branch which is depicted at the bottom of Figure 2, we show the radar data processing chain, which involves multiple steps. Modern automotive FMCW radars transmit several chirps and have multiple receiving antennas to capture the reflections from the surroundings. The echo of a single chirp is used for ranging, and by exploiting the phase shift between consecutive chirps, we also derive the radial velocity of moving objects. Furthermore, we estimate the direction from which a reflection originated by examining the time difference between the moments that a reflection arrived at each receiving antenna. All the ADC samples from sampling the echo of consecutive chirps with multiple antennas constitute the radar data cube, the starting point of the radar object detection pipeline.

Initially, we use the raw data cube to compute the so-called range-Doppler map for each antenna by performing a 2D FFT along the fast and slow time axis. Within the range-Doppler maps, it is relatively easy to detect moving objects. In our work, we derive the noise floor and subsequently use it to filter the cells that contain noise. We set the detection threshold at 10dB over the noise floor. After removing reflections of static objects, we end with several cells in the range-Doppler map corresponding only to moving objects. Finally, we estimate the exact location where each reflection originated by performing another FFT along the antenna axis.

Due to the high resolution of modern automotive radars, a single object's echo will extend into multiple cells within the range-Doppler map. Therefore, the last step in creating a list with moving objects is to group those cells. We utilize the DBSCAN clustering algorithm [10] as it does not require prior knowledge of the number of objects and can also handle spurious detections and noise. Each detection is a tuple that includes information about the range, radial velocity, Direction of Arrival (DoA), and amplitude of the received reflection at a specific cell of the RD map. Therefore, we can extract several features for each cluster of detections, such as its

Authorized licensed use limited to: UNIVERSITY OF TWENTE.. Downloaded on March 13,2024 at 14:56:05 UTC from IEEE Xplore. Restrictions apply.

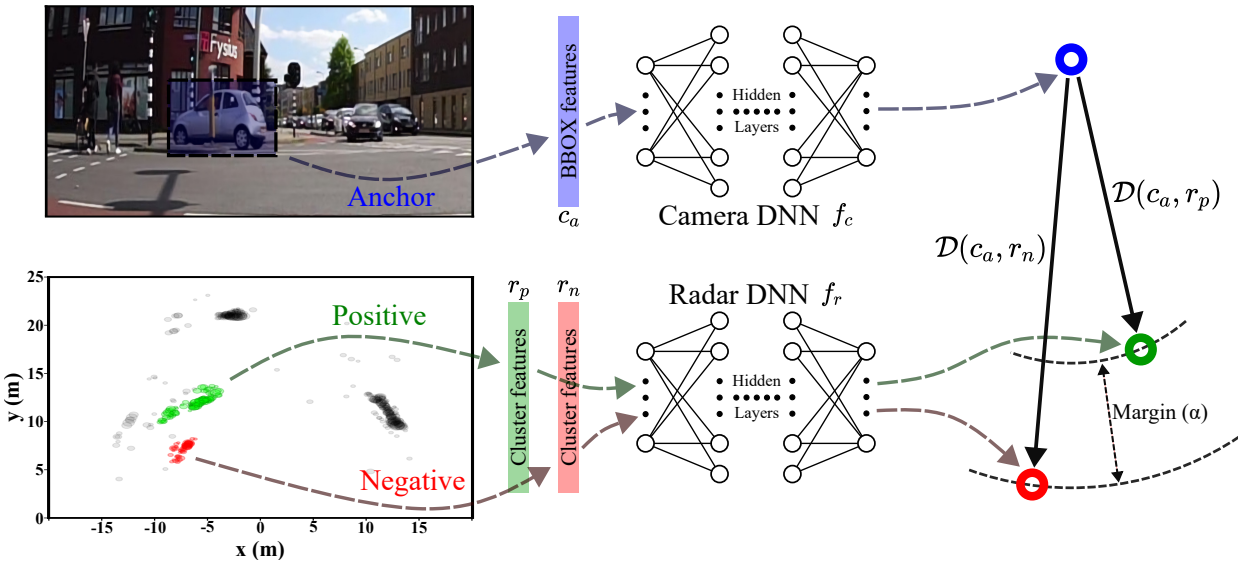

Fig. 3. Jointly training the neural models with triplets. The blue color indicates the anchor input of a triplet, while the green and red color indicates the positive and negative input, respectively.

maximum, minimum, and mean values for the range, velocity, and direction of arrival. All the features extracted from each radar cluster are listed in the left column of Table I.

B. Problem Formulation

Sensor calibration allows the transformation of a point in the image coordinate system to the radar's coordinates system and vice versa. Contrary to point-to-point transformation, our method investigates the similarity at an object level. We do so by taking advantage of the neural networks' flexibility regarding their input, which enables us to provide several features (Table I) for each detection to describe it in more detail.

Let $c_i \in \mathbb{R}^{d_1}$ be the feature vector of the i^{th} detection from the camera sensor. Similarly, $r_j \in \mathbb{R}^{d_2}$ is the feature vector of the j^{th} radar detection. Then for each time-step, $C = [c_1, c_2, \ldots, c_M] \in \mathbb{R}^{d_1 \times M}$ and $R = [r_1, r_2, \ldots, r_N] \in \mathbb{R}^{d_2 \times N}$ are two sets with the feature vectors of detected objects in the visual and radar modalities, respectively. The two sets of detections can be seen on the right side of Figure 2 with colored dashed boxes. Finally, d_1 and d_2 are the dimensions of the object feature vectors that describe camera and radar detections, respectively. As seen in Table I, $d_1 = 7$ and $d_2 = 13$.

Note that the number of detected objects in the image and clusters in the radar can differ. N is the number of detected clusters in the radar point cloud, and M is the number of bounding boxes in an image frame. Given that the detections from each modality are temporally aligned, we want to match the 2D bounding box from the set of camera detections C with its corresponding radar cluster(s) in R.

Therefore, our metric learning based method aims to seek the following radar and camera detection projection functions:

$$f_c : \mathbb{R}^{d_1} \rightarrow \mathbb{R}^d, \quad f_r : \mathbb{R}^{d_2} \rightarrow \mathbb{R}^d,$$

where f_c and f_r are deep neural networks with non-linear activation functions, and d is the dimension of the learned common embedding space. Thus, the two neural models have different input layer sizes. However, they both produce an output vector (embedding) of a similar dimension. We empirically found that setting d to 16 results in the best matching accuracy on our dataset.

In the following subsection, we explain the training procedure to tune the parameters of the two neural networks so that the radar cluster and 2D bounding box features from the same object are projected onto neighboring locations in the shared embedding space.

C. Deep Metric Learning

To optimize the parameters of the neural models, we use the Triplet loss function. This formulation compares a reference input (anchor) to a matching (positive) and non-matching (negative) input. We can describe the loss function using the Euclidean distance function as follows:

$$\mathcal{D}(c, r) = \|f_c(c) - f_r(r)\|_2$$

$$\mathcal{L}(c_a, r_p, r_n) = \max(0,\ \mathcal{D}(c_a, r_p) - \mathcal{D}(c_a, r_n) + \alpha), \quad (1)$$

where c_a is the anchor, r_p and r_n are the positive and negative input, and α is a margin between the positive and negative pair distance. The margin term α is essential for the learning procedure under the triplet loss as it dictates the minimum distance between two non-matching projections. Similarly to [7], we set the margin to 0.2. Finally, $\mathcal{D}$ is the Euclidean distance between a bounding box and a radar cluster projection in the embedding space.

Figure 3 illustrates the training procedure using triplets. The anchor is the bounding box feature vector from the visual object detector, while the positive and negative inputs are radar

Authorized licensed use limited to: UNIVERSITY OF TWENTE.. Downloaded on March 13,2024 at 14:56:05 UTC from IEEE Xplore. Restrictions apply.

cluster features. From (1), it follows that we minimize the loss when the distance between the projection of the bounding box and the radar cluster of the same object is smaller than the distance to the projection of another object's radar cluster.

To provide the triplets, we use a dataset that contains 2D bounding boxes matched with their corresponding radar cluster(s) that belong to the same object. On each epoch of the training procedure, we iterate through every object from our training dataset. We use each object's bounding box as an anchor and its corresponding radar cluster as the positive example. The negative example is a radar cluster of another randomly selected object from the dataset and is different in every epoch.

With the dataset we discuss in the Section IV-A, we jointly trained the neural models for 200 epochs with the Adam optimizer with a learning rate of 10^{-5}. The two neural networks are structurally identical and consist of 3 hidden layers with 128 neurons each. We used the Sigmoid activation function for all hidden neurons, while the output layer of the deep neural networks does not have an activation function.

D. Inference

During inference, we do not provide triplets. Instead, we aim to associate all bounding boxes detected in an image with their corresponding radar cluster using our trained DNNs. Therefore, we first detect all objects in both modalities and construct the camera and radar feature vector lists. Then we feed all the object feature vectors to the corresponding neural model and register its outputs. Finally, we use all the projections to construct a similarity matrix whose entries are the Euclidean distance between all bounding box embeddings and all radar clusters in the shared embedding space.

The similarity matrix has M rows and N columns, where M and N are the numbers of detected objects in the visual and radar modality, respectively (Section III-B). The value for the j^{th} column in the i^{th} row is the Euclidean distance between the embedding of the i^{th} bounding box from the set of detected objects C and the j^{th} radar cluster embedding from the set R of all radar detections at a given time step.

The number of detections from each modality is rarely the same. Therefore, some detections will remain unmatched. In our work, we match radar clusters to the camera detections. The reason is that visual object detectors use non-maximum suppression and generate a single box for each object. On the other hand, radar clusters can include more than one object's reflections, or multiple clusters may exist for the same moving object. Consequently, we match each bounding box embedding vector with the closest radar cluster embedding. The matching function can be described as follows:

$$\mathcal{M}(c_i) = \underset{r_j \in R}{\operatorname{argmin}}\, \mathcal{D}(c_i, r_j),$$

where, $\mathcal{D}$ is the Euclidean distance of two embedding as defined in III-C. We illustrate this process in Figure 1. In this example, the visual object detector correctly detects the two pedestrians crossing the street. However, their reflections are clustered together in the radar data. As a result, their bounding boxes are matched with the same cluster. The other two vehicles are correctly matched with their corresponding cluster. The projection of the remaining radar cluster (blue color) remains unmatched, as the radar captured a vehicle that is not visible in the image crop.

IV. Experiments and Results

In this section, we provide experimental results that we acquired by testing our algorithm on a portion of our dataset. We first provide details on the data we collected to train the neural models and evaluate our method. We then show the simple rule-based algorithm we used as a baseline to compare against our learning-based method. Finally, we provide performance metrics in terms of matching accuracy.

A. Dataset

In the lack of automotive datasets with the annotation needed for our matching algorithm, we collected our own data within the city of Enschede in the Netherlands. Additionally, the access to the raw output of the radar sensor allowed us to choose which features would better describe an object detected in radar modality without limitations. We recorded synchronized image and radar data with a setup consisting of the TEF810x radar transceiver from NXP and a camera with an ultra wide-angle lens. Consequently, all moving objects are captured in both modalities as the sensors have the same field of view, which is 180 degrees.

The recording took place at an intersection with bicycle lanes and vehicle traffic controlled by stoplights. During the recording, we acquired 2840 frames that contained 7140 moving objects. The sampling rate was ten frames per second. We used the first 2090 frames for training and the remaining 750 for testing. The moving objects in the training dataset constitute 70% of the total number of objects. We manually annotated the dataset after detecting all objects in the two modalities with the preprocessing steps we presented in section III-A. The annotation indicates which radar clusters are associated with the bounding box of each moving object for each time step.

B. Baselines

In order to compare the performance of the proposed method, we developed a rule-based model to estimate the position of an object in real-world coordinates. The two sensors are co-located, have the same field of view, and point in the same direction. Therefore, we use the middle pixel coordinate of the bounding box to derive its bearing, which is comparable with a radar cluster's DoA. To infer the range of an object, we use a data-driven approach that takes advantage of the inverse linear relationship between an object's height and distance from the camera.

The exact formulation of our baseline model is as follows:

$$\rho_{bbox} = (\beta h + \gamma)^{-1}$$

$$\theta_{bbox} = \pi - \pi \frac{x}{I_w},$$

Authorized licensed use limited to: UNIVERSITY OF TWENTE.. Downloaded on March 13,2024 at 14:56:05 UTC from IEEE Xplore. Restrictions apply.

where ρ and θ are the estimated radial and angular coordinates of each object, h is the height of its bounding box, β and γ are parameters that we acquire after fitting our model with the least squares method to the training dataset. Finally, x is the middle pixel coordinate of the bounding box, and I_w is the width of the images in pixels, 1920 in this case.

Following estimating an object's position based on its bounding box, we convert its polar into Cartesian coordinates. Hence, we can utilize the Euclidean distance to match each bounding box with its corresponding radar cluster after constructing the similarity matrix as in III-D. However, the metric is the Euclidean distance of two points in the real-world coordinates instead of the distance in the learned embedding space. To calculate the pairwise distance, we define each radar cluster's polar coordinates by considering its mean DoA and minimum range. Then we transform its polar to Cartesian coordinates.

In addition to the rule-based baseline model we described, we also trained a single DNN to estimate the polar coordinates of the objects directly using the bounding box features listed in Table I. In doing so, we can evaluate whether using two neural models and transforming cross-modal features into a joint embedding space is justified. The matching of cross-modal detections is done similarly as with the other baseline.

C. Results

Our method was evaluated and compared against the previously mentioned baselines with a test dataset that consists of 508 frames. This number is less than the total frames we left out for testing, as we only use frames that contain at least two moving objects for our comparison. We measure the mean accuracy over all frames and also in sets of frames with a specific number of objects. The accuracy is defined as the number of bounding boxes correctly matched with their corresponding radar cluster over the total number of visual detections per frame. Additionally, we compute the confidence interval around the mean accuracy.

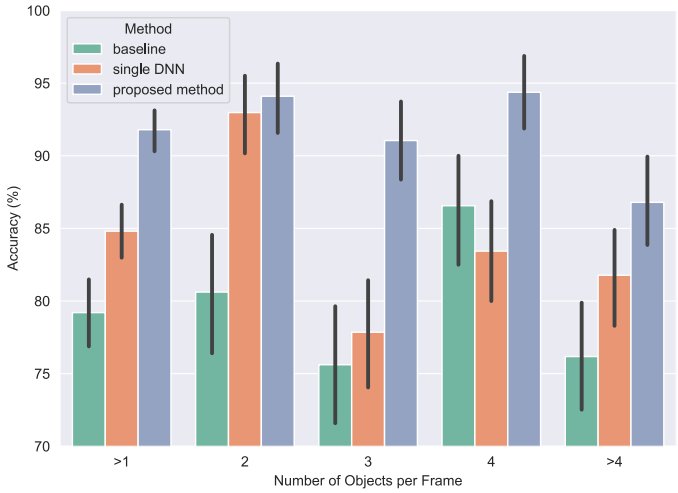

Fig. 4. Performance comparison on the test dataset.

We show the comparison results in Figure 4. Matching objects from the two modalities with the learned metric we propose is consistently more accurate than both baselines. More specifically, our deep metric-based matching algorithm has a mean accuracy of 91,8% on all frames of the test dataset. On the other hand, the rule-based and single DNN baselines have a mean accuracy of 79.2% and 84.8%, respectively.

The main reason that hinders the rule-based baseline algorithms' accuracy is the distance estimate of the bounding boxes, which depends on their height. As the model's parameters are fitted to our dataset that contains more passenger cars than any other class, the model is more accurate in frames containing moving objects with heights similar to a passenger vehicle. However, even the position estimation based on a single DNN that uses multiple features fails to match the performance of our proposed deep metric method.

V. Conclusion

This work proposes a radar and camera object-level association method that relies on two lightweight neural networks which transform heterogeneous features into a shared embedding space. In contrast to point transformations, we can use multiple features to describe moving objects better. We show that jointly training the neural networks with the triplet loss results in a learned space where the distance between object projections is a robust association metric. Using our deep metric, we significantly outperformed detection matching methods that rely on the distance between estimated object positions in a 2D Cartesian plane.

References

[1] 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," *NeurIPS Machine Learning for Autonomous Driving Workshop*, 2019.

[2] 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 *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, 2019, pp. 1–7.

[3] K. Aziz, E. De Greef, M. Rykunov, A. Bourdoux, and H. Sahli, "Radar-camera fusion for road target classification," in *2020 IEEE Radar Conference (RadarConf20)*, 2020, pp. 1–6.

[4] J. Mendez, S. Schoenfeldt, X. Tang, J. Valtl, M. P. Cuellar, and D. P. Morales, "Automatic label creation framework for fmcw radar images using camera data," *IEEE Access*, vol. 9, pp. 83 329–83 339, 2021.

[5] J. Oh, K.-S. Kim, M. Park, and S. Kim, "A comparative study on camera-radar calibration methods," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2018, pp. 1057–1062.

[6] V. E. Liong, J. Lu, Y.-P. Tan, and J. Zhou, "Deep coupled metric learning for cross-modal matching," *IEEE Transactions on Multimedia*, vol. 19, no. 6, pp. 1234–1244, 2017.

[7] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.

[8] X. Dong, B. Zhuang, Y. Mao, and L. Liu, "Radar camera fusion via representation learning in autonomous driving," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 1672–1681.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[10] M. Ester, H. P. Kriegel, J. Sander, and X. Xiaowei, "A density-based algorithm for discovering clusters in large spatial databases with noise," 12 1996. [Online]. Available: https://www.osti.gov/biblio/421283

Authorized licensed use limited to: UNIVERSITY OF TWENTE.. Downloaded on March 13,2024 at 14:56:05 UTC from IEEE Xplore. Restrictions apply.