Weakly Supervised Deep Learning Method for Vulnerable Road User Detection in FMCW Radar

Martin Dimitrievski Ivana Shopovska David Van Hamme Peter Veelaert Wilfried Philips

TELIN-IPI, Ghent University - imec
Sint-Pietersnieuwstraat 41
B-9000 Gent, Belgium

***Abstract*—Millimeter-wave radar is currently the most effective automotive sensor capable of all-weather perception. In order to detect Vulnerable Road Users (VRUs) in cluttered radar data, it is necessary to model the time-frequency signal patterns of human motion, i.e. the micro-Doppler signature. In this paper we propose a spatio-temporal Convolutional Neural Network (CNN) capable of detecting VRUs in cluttered radar data. The main contribution is a weakly supervised training method which uses abundant, automatically generated labels from camera and lidar for training the model. The input to the network is a tensor of temporally concatenated range-azimuth-Doppler arrays, while the ground truth is an occupancy grid formed by objects detected jointly in-camera images and lidar. Lidar provides accurate ranging ground truth, while camera information helps distinguish between VRUs and background. Experimental evaluation shows that the CNN model has superior detection performance compared to classical techniques. Moreover, the model trained with imperfect, weak supervision labels outperforms the one trained with a limited number of perfect, hand-annotated labels. Finally, the proposed method has excellent scalability due to the low cost of automatic annotation.**

***Index Terms*—deep learning,radar, weakly supervised, VRU detection**

I. Introduction

Frequency modulated continuous wave (FMCW) radar has the capability to directly measure an object's range and radial velocity. Owning to these unique properties, radars have been installed in numerous land, maritime and airborne platforms for the tasks of object detection and tracking. While classical signal processing can be applied efficiently to detect large objects, discriminating people (referred to as VRUs throughout the paper) from clutter in traffic environments remains a difficult task. This is manly due to the fact that VRUs are poor radar energy reflectors and they move slowly relative to the static environment. Additionally, the effects of multipath propagation of radar signals are difficult to model explicitly due to the unknown and ever changing scene geometry. Yet, detecting moving people in radar data can be performed based on the unique pattern of motion of the human body. Specifically, a pedestrian or a cyclist exhibits an oscillatory motion of the limbs which is commonly referred to as a micro-Doppler signature. It is therefore possible to fit a model of the kinematics of the human gait to measured data and distinguish a person from other objects. However, radar signal is often distorted by the phenomenon of multi-path propagation, i.e. returns from multiple sources interfering with the signal reflected from the object of interest. Without prior knowledge of the scene geometry, detecting objects of interest becomes very difficult using classical forward models. The scientific community agrees that radar signal processing has to be extended to concepts from machine learning and pattern recognition in order to keep radar in the leading edge of remote sensing [1].

A deep Convoltuional Neural Network (CNN) acts as a universal function approximator with the capability to learn any function given enough training evidence. This makes the VRU detection problem inherently suitable for deep learning since the complexity of the input space is difficult to interpret, but capturing large amounts of radar measurements is relatively easy. Unfortunately, annotated raw radar datasets are currently not publicly available and expert knowledge for making manual annotations is not readily available. In this paper we propose to train a radar CNN for VRU detection by weak supervision from calibrated camera and lidar. By removing the human expert from the annotation loop we are able to completely automate the training process. The radar CNN parameters are thus optimized using object positions and their existence uncertainties in a weighted cross-entropy loss function. Essentially, we train the radar network to perform both object detection with lidar ranging precision and classification at camera-level accuracy. The trained model can then be deployed on a robot that operates without the need for an expensive lidar. Although our automatically generated labels are imperfect, we experimentally show that this strategy has great benefits for cost-effective training, resulting in a significantly better detection rates. By utilizing the abundance of automatically labeled data, we were able to train a model with performance beyond what is practically achievable using manual annotations.

The remaining of the paper is organized as follows: a brief overview of the relevant literature is given in section §II, where we point to the fact that several deep learning methods for detecting various objects in Radar data already exist. In section §III we present details about the main contribution while in section §IV we present the experimental evidence

from training networks of the same architecture in a fully supervised and weakly supervised manner. Finally, in section §V we present some remarks about the limitations of the proposed method and make suggestions for further improvements.

II. Related work

Classical methods such as the Constant False Alarm Rate (CFAR)[2] perform moving target detection in radar signal by determining whether a target exists in the clutter or noise background. Existing CFAR detection procedures are commonly performed using sliding windows, from which, the parameters of the hypothesized model are estimated, and the data available in the reference window are employed to compute the decision threshold. CFAR offers reliable detection of moving targets, however, further processing on these detections is needed for classification. In [3], the authors present one typical example of detecting the motion of people using hand crafted range and Doppler features. Their method is built on analyzing the radar waveform design, i.e. the expected characteristics of the return signal given the known radiation pattern and the most likely object motion features. A comparative study [4] analyses the performance between random forests and LSTM, see [5], network for classification of cars, pedestrians, groups, bikes and trucks. Backed by large scale experiments, their conclusion is that the difference between LSTM and random forest is surprisingly small (0.884 vs. 0.871 F1 score). They also found that the performance of the LSTM network, in particular, is highly sensitive to the amount of training samples, a motivation which drives us towards training with large dataset and weak supervision.

A semantical radar grid building algorithm is presented in [6]. Authors rely on 4 radars, whose observations are first registered, to classify regions containing cars and other objects. In this paper a shallow fully convolutional neural network was used to classify input occupancy grid cells into classes of objects. In contrast to this approach, we're interested in instantaneous VRU detection and operate on a short time window of only 5 consecutive radar measurements which are not registered. In [7], authors present a semi-supervised deep radar detector operating on 4D dense radar data. The method splits the input dimensions by applying two independent CNNs that process in the range-Doppler and elevation-azimuth dimensions respectively. A weakness of this method is that radar dimensions are only combined in late feature space, thus the potential of inter-dimensional dependencies is lost from the start. Since we are only interested in detecting VRUs, we can discard any unnecessary Doppler data by pre-processing steps, and retain a complete 4D radar space as input. Authors of the paper [8] propose a CNN object detection and 3D estimation based on the U-Net architecture. They use a coupled Radar and Camera sensor to prepare a set of training samples for a radar CNN which determines the presence or absence of a car in the radar signal. This method uses a 3D network architecture, where the input tensor consists of radar range, velocity and receiver channel information, and the output consists of 3

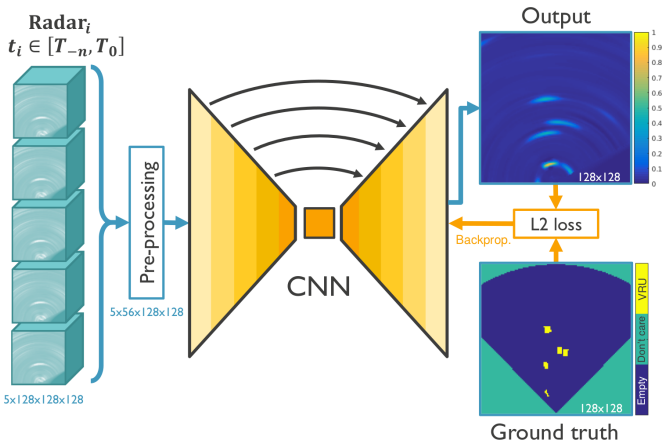

Figure 1. Diagram of the proposed radar detector. Radar arrays are pre-processed and fed into a multi-resolution segmentation CNN producing a probability of occupancy grid. The networks optimizes a weighted cross-entropy function between it's output and automatically generated ground truth from camera and lidar.

layers: a binary probability of occupancy and two image plane coordinates. The main drawback of this method is that its training protocol is limited to cars in the image plane and the authors do not provide extensive evaluation for cluttered environments.

In [9] authors propose a hybrid radar detection system consisting of initial target detection by classical processing followed by radar target classification network. The network operates on cropped range-azimuth-Doppler radar tensors extracted around the initial radar targets and outputs a class label and score for the categories car, person and cyclist. Finally, they apply clustering in order to group similarly classified targets into complete objects. This method was evaluated on a real-world dataset using automatically annotated ground truth from matched camera and stereo-depth sensors. Even though authors report promising results, this method relies on single time integration radar cubes, therefore overlooking important micro-Doppler cues needed for classifying VRUs.

Finally, a comprehensive analysis of applying deep learning to radar signals is presented in [10]. The authors of this paper propose a deep learning method for vehicle detection in bird's eye view using Range-Azimuth-Doppler tensors. Interestingly, the method doesn't truly work with the full 3D radar data, rather it computes three image-like inputs by collapsing each radar dimension respectively. This paper also proposes a semi-automated annotation framework based on a 64 beam lidar sensor, however manual human correction was needed to obtain ground truth. As a result of an ablation study, the following conclusions were reached: best performance is achieved by operating in the native polar coordinates and applying a Cartesian transformation on the latent features , second: incorporating Doppler information using their proposed model has marginal benefits and third: exploiting the temporal dependency with a LSTM cell has marginal benefits . A potential weakness of this method is the recurring loss of micro-Doppler information due to the collapse of dimensions in the pre-processing.

III. Proposed method

Our goal is to train a radar neural network for VRU detection on the ground plane using weakly supervised deep learning. To that end, an abundance of imperfect training samples will be provided by automatically matching camera and lidar objects. The radar network will therefore be trained to both classify and estimate the position of VRUs at the same time. Formally, the radar CNN model $f_{CNN}()$ computes an estimate Y of the occupancy grid of the environment M, given series of consecutive, pre-processed, radar measurements Z_r, $Y = f_{CNN}(f_{pre}(Z_r))$. Each cell of the output grid $y_{\rho,\theta}$, figure 1- top right, contains the probability that the region is occupied by a VRU. Weak supervision is provided from camera and lidar sensor measurements: Z_c, Z_l which we transform into a ground truth estimate $\hat{M}$, figure 1- bottom right. Finally, detection of object's centers is easily performed by estimating local peaks in the output occupancy grid Y.

A. Pre-processing

Typical radar data streams are usually represented as dense 3D arrays containing time integrated Range-Azimuth-Doppler signals, or 4D arrays if the radar also measures elevation. Since most of the radar contains little information, feeding the complete array to a CNN is sub-optimal and computationally expensive. We therefore develop a pre-processing algorithm $f_{pre}()$ built on domain specific knowledge which results in data reduction at no loss of information. Knowing that the mean frequency of human gait is around to 1Hz, while recreational cyclists on average pedal at a cadence of roughly 60RPM, we deem that only a half-period of this motion is sufficient to extract its characteristic patterns. We therefore concatenate radar measurements spanning the time period from $500ms$ in the past until the present. At time $t = 0$ we have the current and past radar arrays (range-azimuth-Doppler cubes) $Z_r = \{\mathbf{R}_{-4},\mathbf{R}_{-3},\mathbf{R}_{-2},\mathbf{R}_{-1},\mathbf{R}_0\}$ where each one is captured at an interval of $100ms$.

In order to reduce the effect of range dependent signal decay, we apply a standard pre-processing step from classical CFAR [11], i.e. estimation of the normalized power $\mathbf{P}$. This has the effect that newly computed values are distance independent and proportional to the local signal to noise characteristics. The underlying structure is therefore easier to interpret by the neural network. The normalized power for a one dimensional signal can be computed as:

$$\mathbf{P}(i) = \mathbf{R}(i)^2 \left[\frac{1}{2N} \sum_{l=G+1}^{G+N} \mathbf{R}(i+l)^2 + \mathbf{R}(i-l)^2 \right]^{-1}, \tag{1}$$

where N is the number of cells used for estimating the noise floor and G is the number of guard cells which are skipped in order to avoid sampling the object under test. The procedure can easily be extended in 3D and efficiently computed by a 3D convolution.

In moving radar systems, a typical artifact is the apparent shift of structure proportional to the ego-velocity $-v_{ego}$ and the cosine of the azimuth. We remove this velocity vector from the captured data by transforming it into a new, ego-velocity independent space. Theoretically, ego-velocity can be estimated from the Doppler velocity of static objects in front of the radar. This comes from the simple fact that the perceived radial velocity of static objects along longitudinal axis is directly proportional to the ego-velocity. However, in practice the objects right in front of the radar are rarely static which causes such ego-velocity estimation to fail. Thus, we estimate the ego-velocity by approximating it from the radial velocities in the circular sector $\theta_0 \in [-30°, 30°]$ which is more likely to contain static objects. The ego velocity estimate $\hat{v}_{ego}$ then is the velocity of the Doppler slice containing the highest normalized power in this circular sector:

$$\hat{v}_{ego} = -\operatorname*{argmax}_{v} \sum_{\rho=0}^{\rho_{max}} \sum_{\theta=-30°}^{30°} \mathbf{P}_{\rho,\theta,v}. \tag{2}$$

Each individual radar cube is thus corrected for ego-motion by shifting along the Doppler dimension so that the data is centered around the estimated bin: $\mathbf{P}_{\rho,\theta,v} \leftarrow \mathbf{P}_{\rho,\theta,v-\hat{v}_{ego}}$. Lastly, Doppler slices of velocities that far exceed normal VRU velocities are also removed. We make sure that discarding Doppler slices will not impact detection performance by taking a wide margin of $\pm 3ms^{-1}$, knowing that people's body parts do not move much faster than their mean velocity. This helps us reduce the size of the CNN input $\mathbf{P}_{\rho,\theta,v}$ and thus lessen the load on the GPU for training.

B. Radar CNN architecture

The task of detecting moving VRUs practically consists of localization and classification. While localization can be done relatively effectively using single-frame processing, VRU classification in radar necessitates the use of temporal information. A joint detector-classifier therefore requires both a wide spatial receptive field, as well as significant depth in the time-Doppler dimension. The former is needed for learning to separate targets from each other and from multi-path reflections, while the later is essential for gait classification. The combination of spatial layers and memory modules has the disadvantage of costly training, where data sequences must be processed as time series and GPU resources cannot be fully utilized. We therefore choose to concatenate five consecutive radar cubes along a common time-Doppler dimension and use a 2D U-Net architecture.

Following pre-processing, our information dense tensor is fed to the contracting head of the U-Net where a series of convolutions and max-pooling operators reduce the spatial information into a more dense feature space. The bridge of the network, containing two Fully Connected (FC) layers, then classifies the presence of moving VRUs. Up-sampling is performed in expansion blocks using the dense FC features and high resolution information from the contracting blocks via skip and concatenation layers. Finally, a sigmoid activation function is used to map the network output to predict the probability for occupancy of a VRU at each range-azimuth cell. For training, we use a per-sample and per-class weighted, two-class cross-entropy loss function. Per-sample weights account for the varying confidence in our weakly

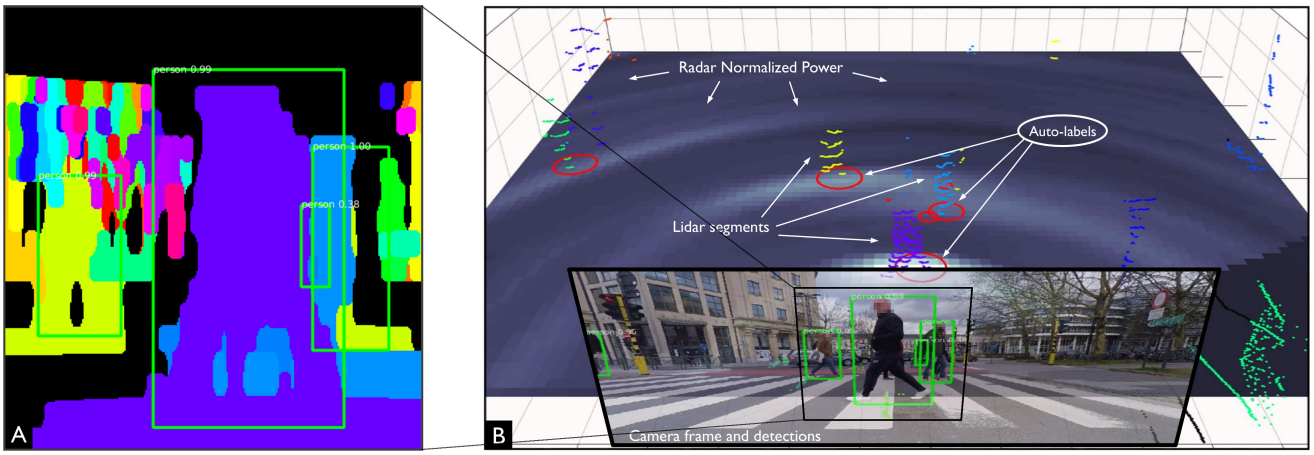

Figure 2. Automatic label generation for weakly supervised training of radar CNN. A: instance segmentation masks from lidar projected on the camera image are matched to Faster R-CNN BBs, B: 3D scene visualization of the input sensor data and the computed weak supervision training labels.

supervised ground truth at the specific cell, while per-class weights adjust the desired sensitivity of the output.

C. Automatic annotation using Camera and Lidar

Annotating data is a labor intensive and expensive process. This task is especially difficult when labeling raw radar data which is non-intuitive to the untrained eye. In order to reduce the costs for obtaining a dataset adequate for deep learning, in this section we will show how to automatically compute a ground truth estimate $\hat{M}$ using external sensor measurements, namely camera Z_c and lidar Z_l. The proposed estimates can be used as weak training labels in the in the form of an occupancy grid map estimates as defined in [12].

Formally, our weak supervision grid $\hat{M}$ is an estimate of the true occupancy grid M defined by the conditional probability of occupancy of cell $m_{\rho,\theta}$ given the positions of true VRUs $p\left(m_{\rho,\theta}\left|X\right.\right)$; $X:\{\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_k\}$; $0\leq k$; $\mathbf{x}_i=\begin{bmatrix}\rho_i & \theta_i\end{bmatrix}^T$. We compute this estimate as the conditional probability of occupancy $p\left(\hat{m}_{\rho,\theta}\left|Z_c,Z_l\right.\right)$, using the inverse sensor model given sensor observations Z_c and Z_l. On the image plane, we compute the set Z_c consisting of n Bounding Boxes (BB) by running the Faster R-CNN object detector [13], $Z_c=\{\mathbf{c}_1,\mathbf{c}_2,...,\mathbf{c}_i\}$. The output of the detector is a vector of center image coordinates, BB width and height and detection score: $\mathbf{c}_i=\begin{bmatrix}r_i & c_i & w_i & h_i & s_i\end{bmatrix}^T$. The lidar observation Z_l is a set of objects which we compute by segmenting the point cloud into disjoint objects, $Z_l=\{\mathbf{l}_1,\mathbf{l}_2,...,\mathbf{l}_j\}$ using [14]. Each object is a vector consisting of its ground plane center and a unique instance identifier: $\mathbf{l}_j=\begin{bmatrix}\rho_j & \theta_j & \text{ID}\end{bmatrix}^T$.

Assuming that no grid cell can be occupied by more than one VRU, the true map M can be approximated from the set of true VRU positions X using a kernel density function in 2-D space. For simplicity, we use the 2-D Dirac delta:

$$M=\iint p\left(m_{\rho,\theta}\left|X\right.\right)d\rho d\theta\approx\sum_{i=1}^{|X|}\delta\left(\rho-\rho_i,\theta-\theta_i\right). \quad (3)$$

Our weak training supervisor uses a set of estimated object positions (labels): $\hat{X}:\{\hat{\mathbf{x}}_1,\hat{\mathbf{x}}_2,...,\hat{\mathbf{x}}_m\}$, each representing a VRU $\mathbf{x}_i$ by its ground position $\begin{bmatrix}\rho_i & \theta_i\end{bmatrix}^T$ and the belief of it's existence s_i. The weak supervision mask $\hat{M}$ can thus be expressed through the probability that a ground plane cell $\hat{m}_{xy}$ is occupied by object $\hat{\mathbf{x}}_i$:

$$\hat{M}=\iint p\left(\hat{m}_{xy}\left|\hat{\mathbf{x}}_i\right.\right)d\rho d\theta;\ \hat{\mathbf{x}}_i\in\hat{X}=f_{match}\left(Z_c,Z_l\right), \quad (4)$$

where each object $\hat{\mathbf{x}}_i$ is computed by matching detections in the camera and lidar. In practice, finding likely correspondences between BBs in Z_c and lidar objects in Z_l is not trivial. From the example on figure 2 it is clear that a one-to-one mapping between BBs and instance segmentation masks can be ambiguous i.e. the matching task $f_{match}:Z_c\rightarrow Z_r$ is non-injective and non-surjective. A BB can often contain instance segmentation masks from multiple lidar objects, and there are many lidar objects that are not VRUs. One way to cope with this complexity is to limit the matching only to lidar objects projected within the boundaries of the camera bounding box under test $\mathbf{c}_i$. The simplest solution for estimating the position of $\hat{\mathbf{x}}_i$ is to match the bounding box $\mathbf{c}_i$ with the most likely lidar segment $\mathbf{l}_{\text{MLE}}$. Thus each object $\hat{\mathbf{x}}_i$ in (4) will be defined by the range and azimuth of the maximum likelihood estimate (MLE) lidar object and the camera object score s_i. For a single, non-occluded person we model the position of $\hat{\mathbf{x}}_i$ as the uni-modal distribution:

$$\hat{\mathbf{x}}_i\sim\begin{cases}\delta\left(\rho-\rho_{\text{MLE}},\theta-\theta_{\text{MLE}}\right), & \text{if } s_i\geq\tau,\\ 0 & \text{otherwise,}\end{cases} \quad (5)$$

where δ is the Dirac delta function expressed in polar coordinates. The existence score s_i, as provided by Faster R-CNN, is used as a latent variable controlling the existence of this PDF. For notational simplicity we will assume that each object $\hat{\mathbf{x}}_i$ is always existent skipping the degenerate case in the remainder of the analysis. Practically, the most likely lidar object $\mathbf{l}_{\text{MLE}}$ within a BB $\mathbf{c}_i$ is the one with the highest overlap:

$$\mathbf{l}_{\text{MLE}}=\underset{j}{\text{argmax}}\left[IoU\left(\mathbf{c}_i,\mathbf{l}'_j\right)\right], \quad (6)$$

by means of the Jaccard Index between the BB and image projections $\mathbf{l}'_j$ of $\mathbf{l}_j$.

Even though the MLE solution can be used as training supervision, see (IV-A), it fails to account for the tails of the distribution which are especially heavy in ambiguous cases of uncertain matching. We therefore propose a more robust solution of computing $\hat{M}$ through the use of a soft association function $f_{match} : Z_c \rightarrow Z_r$ which allows each BB to be matched to multiple lidar blobs and vice versa. The effect of this soft association is that the PDF of each VRU can be spread over multiple modes relative to the quality of camera and lidar matching. We consider each object's $\hat{\mathbf{x}}_i$ probability density function to be multi-modal and localized at discrete ground plane cells. Thus, this PDF is a weighted sum of L peaks located at the centers of mass of each lidar segment $\mathbf{l}_j$ that is visible within the BB $\mathbf{c}_i$:

$$\hat{\mathbf{x}}_i \sim \sum_{j=1}^{|Z_l|} w^j \delta\left(\rho - \rho_j, \theta - \theta_j\right), \tag{7}$$

with weights w_i^j proportional to the matching quality:

$$w_i^j = \frac{\mathrm{IoU}\left(\mathbf{c}_i, \mathbf{l}'_j\right)}{\sum_j^{\mathbf{l}'_j \in \mathbf{c}_i} \mathrm{IoU}\left(\mathbf{c}_i, \mathbf{l}'_j\right)}, \tag{8}$$

where the weights are normalized by the total IoU score for all segmentation masks within the respective bounding box. BBs dominated by a single segmentation mask (e.g. the large, central, purple segment in figure 2) will be modeled as a single peak in the occupancy map. On the other hand, ambiguous objects such as the person in the BB on the right in figure 2 are modeled by a multi-modal PDF with peaks proportional to the uncertainty of detection and matching. The plot B on figure 2 shows these computed PDFs as red circles where a bigger radius is proportional to a higher certainty about the peak's position. By plugging (7) into (4) we arrive at the final form of our approximated occupancy map $\hat{M}$ used during training:

$$\hat{M} = \sum_{i=1}^{|\hat{X}|} \sum_{j=1}^{|Z_l|} w_{ij} \delta\left(\rho - \rho_{ij}, \theta - \theta_{ij}\right). \tag{9}$$

It is clear that the quality of $\hat{M}$ will be influenced by two factors: firstly, the performance of the image object detector, i.e. how well s_i explains the existence of a VRU, thus dictating the cardinality of the set $\hat{X}$, and secondly, the quality of matching $IoU\left(\mathbf{c}_i, \mathbf{l}'_j\right)$ which measures how well the camera and lidar are calibrated, synchronized and how much occlusion is present in the area.

D. Loss function and regularization

We perform training of the CNN parameters by weak supervision from the estimated occupancy grid $\hat{M}$ using a per-sample and per-class weighted cross-entropy function. Specifically, we use two categorical class labels ($C = 2$), one encoding the empty space, and the other the space occupied by VRUs. Thus, the loss function between a network output Y and ground truth label $\hat{M}$ is defined as the weighted sum:

$$loss\left(Y, \hat{M}\right) = \sum_{\rho} \sum_{\theta} w_{\rho,\theta} l\left(y_{\rho,\theta}, \hat{m}_{\rho,\theta}\right), \tag{10}$$

where $l()$ is the cross-entropy

$$l\left(p, q\right) = -\sum_{i=1}^{C} p\left(i\right) \log q\left(i\right), \tag{11}$$

and the weights $w_{\rho,\theta}$ incorporate the object detection score s_i from Faster R-CNN:

$$w_{\rho,\theta} = \begin{cases} \alpha_{pos} s_i & \forall \hat{\mathbf{x}}_i : [\rho, \theta] = [\rho_i, \theta_i], \\ \alpha_{neg} & \text{otherwise.} \end{cases}$$

and use the parameters $\{\alpha_{pos}, \alpha_{neg}\}$ to adjust the detector specificity by reducing class imbalance. In practice, ${}^{\alpha_{pos}}/_{\alpha_{neg}} \gg 1$ which heavily penalizes errors in grid cells containing people compared to errors in empty cells. Also, due to s_i, (10) penalizes more heavily errors in cells which are believed to contain VRUs. The effect of this weighting scheme is two-fold: firstly, network coefficients will adapt to produce strong activations at grid cells matched to highly confident detections from Faster R-CNN, and secondly: there will be strong activations at grid locations with high quality matching between the camera and lidar objects.

We use three different regularization techniques which help with parameter stability and minimize over-fitting. Firstly, the network architecture design itself includes dropout and batch-normalization layers. Regularized network parameters become more robust to perturbations in the input data and are able to converge faster due to the reduction of internal covariance shift [15]. Secondly, we apply a multi-epoch training protocol using the (Adaptive Moment Estimation) ADAM optimizer [16] and apply weight decay of $5 \cdot 10^{-3}$ to all network parameters. We reduce the global learning rate by a factor of 10^{-1} every 10 epochs starting from 10^{-3}. Lastly, we apply a realistic data augmentation technique which randomly flips and rotates the input tensor along the longitudinal axis. Each radar field is flipped along its longitudinal axis, $\theta = 0$, according to the Bernoulli distribution: $\theta' \sim -1^k \theta; p\left(k\right) = 0.5$, and random rotation $\triangle\theta$ is applied along the vertical Z-axis ($\rho = 0$) such that: $\theta'' = \theta' + \triangle\theta; \triangle\theta \sim \mathcal{N}\left(\theta', 64\right)$. These 2D rotations and reflections are Euclidean plane isometries and thus preserve geometrical properties such as lengths and reflection angles. Augmenting the dataset in this way creates an abundance of new realistic samples that retain the effects of multi-path propagation, the very same artifact we want our network to learn to suppress.

IV. Experiments and implementation details

A. Dataset and evaluation protocol

For the purpose of evaluating the proposed method, we captured and annotated a real-world dataset containing multiple scenarios with various traffic conditions and complexity.

Table I
PERFORMANCE EVALUATION RESULTS OF RADAR DETECTORS ON A CONTENT-INDEPENDENT TEST SET OF 489 FRAMES AND 1292 VRUS.

Method	AP	Recall at 0.5 prec.	Training information
RADAR raw	0.150	0%	Empirically optimized
RADAR non-static	0.302	26%	
CFAR (NMS on **P**)	0.439	53%	
CNN-manual	0.513	57%	Supervised: 1351 frames, 3917 VRUs
CNN-auto *	0.556	61%	Weakly supervised: MLE(5): 6955 frames, 7781 VRUs.
CNN-auto **	**0.600**	**69%**	Weakly supervised: multi-modal(7): 6955 frames, 30452 VRUs

In these experiments, the ego-vehicle is driving on public roads in a dense European city center, where multiple VRUs are encountered on the sidewalks and on marked and unmarked crossing zones. The data covers situations from poorly lit environments (20% of the data) to well lit sequences captured in daylight. The data capturing ego-vehicle is equipped with a calibrated and synchronized sensor array consisting of an RGB camera (GoPro Hero 6 Black), a 77GHz FMCW radar (Texas Instruments AWR1243) and a 3D lidar (Velodyne VLP-16). Data was captured and timestamped on a Linux laptop on board the ego-vehicle.

In order to avoid cross-contamination of training and test data, we split the recording into two content-independent parts by selecting data captured at different time and in different parts of the city. Four human annotators were tasked to label both the training and testing part, creating high quality labels by looking for people visible in both the RGB image and the lidar point cloud. The human annotators were able to accurately label 1840 frames generating 4988 VRU positions which took them about 30 hours to complete. At the same time, by running our fully automated annotation tool over the training part we labeled a total of 6955 frames containing 30452 VRUs that will be used for weak supervision training. Note that the high amount of auto-labels stems from the multi-modal definition in (7). Labels consist of the 2D ground plane position of each person, relative to the ego-vehicle origin. Due to the limited resolution of the available VLP-16 lidar, the areas beyond 20m and outside of the view of the camera are considered as "don't care" regions where we ignore detections.

In a series of experiments we applied different supervision learning methods to train the same CNN architecture. Firstly, by using supervision from human annotated labels and then using labels computed automatically. Our hypothesis is that even though the auto-labels are imperfect, their abundance can be beneficial for training a better performing detector. In that regard, the control control CNN (CNN-manual) was trained using annotations from the 10 hand labeled training training sequences. Then, a weakly supervised CNN (CNN-auto *) was trained using the MLE solutions from equation (5) computed from the training part of the dataset,

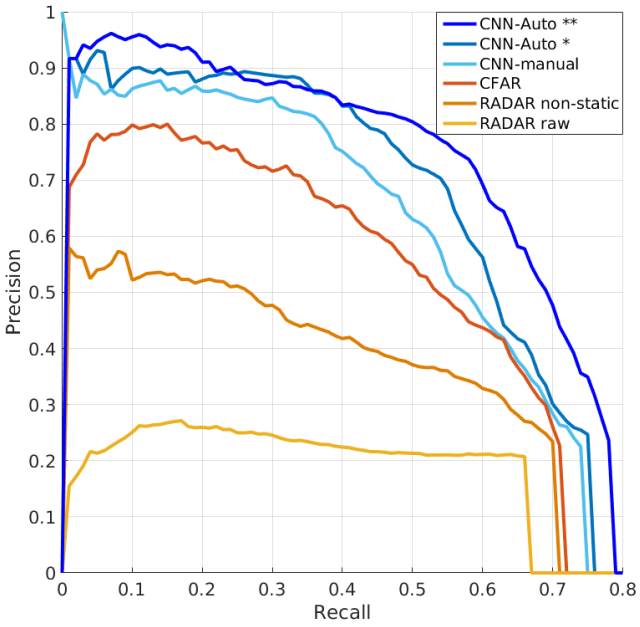

Figure 3. Precision-recall curves for the test set (higher area under curve is better). The results of the proposed method "CNN auto" are compared with a control network "CNN manual" and other classical peak finding techniques.

and finally, we trained a second weakly supervised CNN (CNN-auto **) using the multi-modal formulation from equation (7). After a fixed number of training epochs, each model was evaluated on the content-independent test set. We measure detection performance by varying the detection threshold and computing the proportion of true positive, false positive and false negative samples. To that end, a non-maximum suppression (NMS) algorithm finds peaks in the raw radar signal or the CNN output, which we then match to annotated objects. We use a 5×7 neighborhood of range-azimuth cells which equates to an area of $1.46m \times 7.04$deg in the physical world. A detection is considered true positive if falls within a gate of $3m$ around a ground truth object, while multiple matches within the same gate are not allowed. Based on these statistics, a precision and recall curve is generated for each detector. Finally, we computed the average precision (AP) by taking the mean of the precision sampled at uniformly spaced recall points.

We present a summary of the results in table I and the computed precision-recall plots on figure 3. The detection performance of the proposed method (in shades of blue) is compared to four other algorithms (yellow, orange and red lines). The weakly supervised CNN-Auto** significantly outperforms all other methods in terms of Average Precision (AP). We report an increase of 8.7% AP over the control CNN which was trained using manually annotated training data. Moreover, by allowing our method to learn the uncertainties about detection and matching in the automatically generated labels, equation (7), brings additional performance benefit of 4.4% over training by using the most likely camera-lidar matches, equation (5). Finally, compared to classical peak finding, the proposed CNN-auto** outperforms CFAR (yellow curve in figure 3) by 16.1%. Naïve detection algorithms, such as peak finding in the raw signal and in the

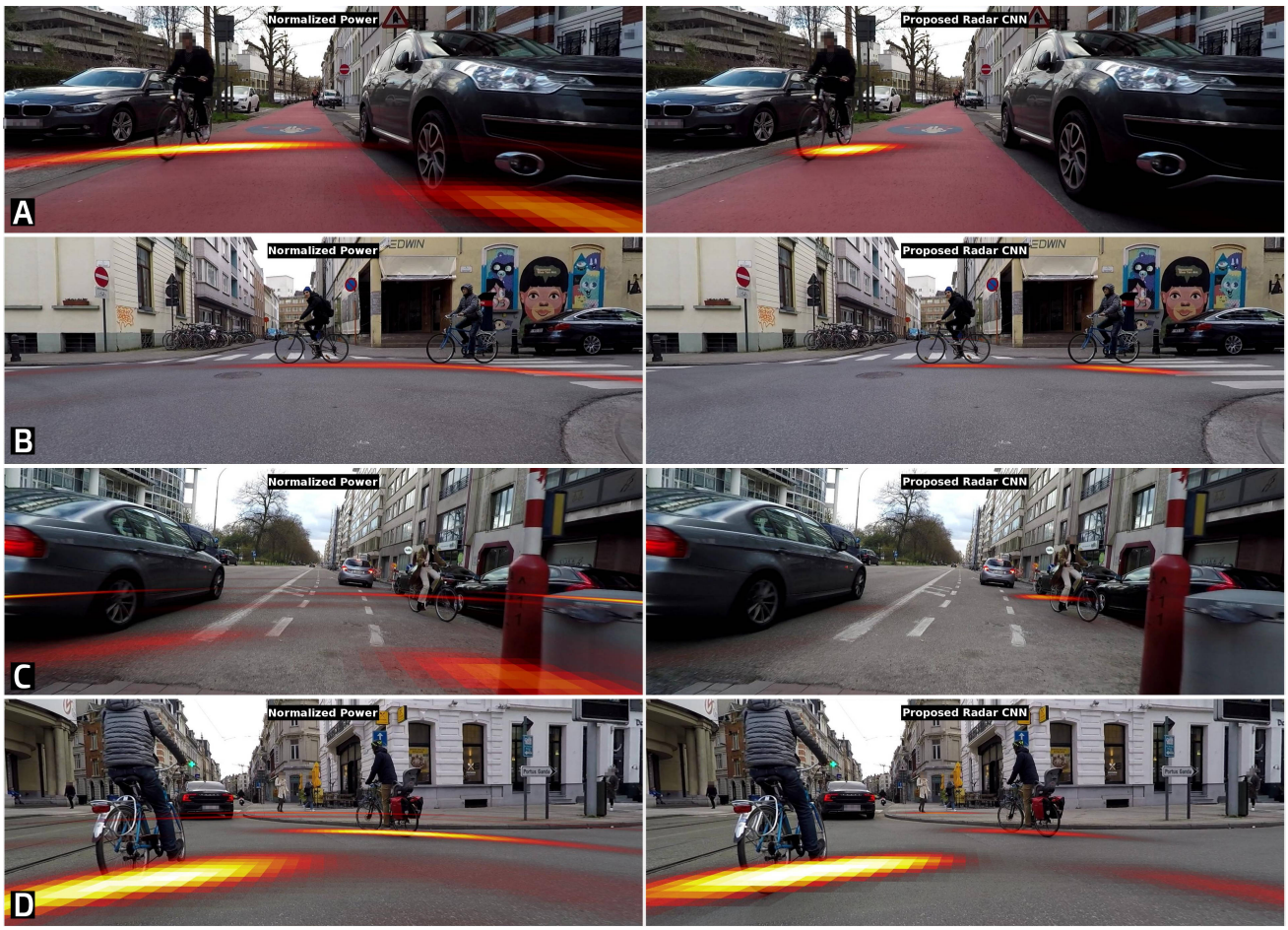

Figure 4. Example frames comparing the qualitative difference of the input radar signal (left column) and the CNN output (right column). A: suppression of clutter from vehicles; B: spatial separation of VRUs in close proximity; C: suppression of clutter from infrastructure; D: suppression of moving vehicles.

moving data, compare unfavorably on our dataset. On figure 4 we present typical cases of operation of the proposed method where we compare the input tensor (left column) to the CNN output (right column). In order to visualize the 280 channels of the input tensor, we collapse it to a 2D array by taking the maximum along the time-Doppler dimension and project it on the respective camera frame. On the right we project the CNN output on the camera image, i.e. the probability of occupancy of a VRU. From these typical examples it is clear that the network output dramatically reduces false positives while at the same time improving the object localization.

B. Implementation details

A FMCW radar frame representing signal strength over range-azimuth-Doppler space is represented as a dense 3-D array. We set the programmable sampling frequency of the TI AWR1243 radar to $10Hz$. Range and Doppler information is encoded in 128 equally spaced bins spanning $0m$ to $46.72m$ and $\pm 13.8m/s$ respectively, while azimuth is encoded in 16 equally spaced bins over the range of $\pm\pi/2$. Power-normalization, (1), is performed by computing the local Signal to Noise Ratio (SNR) T [11] using a 3-D convolution of the input radar array $\mathbf{R}$ with a 3D filter mask. We used a mask with support size of $[15, 11, 1]$ and a guard size of $[5, 3, 0]$ for range, azimuth and velocity respectively. After estimating and correcting for ego-motion, from the original 128 Doppler bins, we discard 5 Doppler bins encoding low velocities: $|v| \leq v_{ego}+2Km/h$ and 34 high velocity Doppler bins: $|v| \geq v_{ego} + 23Km/h$. Each training tensor is created by concatenating 5 pre-processed radar arrays that span over a time interval of $500ms$ by skipping every other frame. The final CNN input tensor consisting of 280 time-Doppler slices.

The network architecture is a U-Net [17] with 5 contraction blocks, a Fully Connected (FC) bridge and 5 expansion blocks. The network outputs a 2D occupancy grid in polar coordinates with spatial resolution matching the one from the input data. Every contraction block applies 3 groups of convolution, batch-normalization (BN), ReLU and a dropout layer followed by a max pooling operator at the end. At the bridge, the input tensor is reduced to spatial resolution 1×1 and 512 dimensional feature space which is input to two fully connected layers. The expansion blocks are built as inverse convolutions (ConvT) initialized to perform up-sampling with linear interpolation, followed by BN and ReLu. In order to preserve high resolution details, up-sampled results are concatenated with feature maps from the respective contracting blocks. Expansion blocks are exempt from dropouts since their task is data unpacking and mixing. Fastest convergence was achieved by training both

CNN-manual and CNN-auto using weighted cross-entropy loss in conjunction with the ADAM optimizer. In all our experiments, we apply early stopping, i.e. we terminate the training once the validation loss starts increasing. Generally, we observed model convergence after ~ 15 epochs or after 110K back-propagations. We used variable training batch size (BS) starting from $BS = 1$ in the first epoch to $BS = 16$ for the remaining. All of these design choices have a direct impact on either the convergence speed or the loss value at convergence. We note that each hyper-parameter value has been chosen meticulously by running control experiments which are outside of the scope of this analysis.

V. Conclusion

In this paper we presented a spatio-temporal CNN for detection of VRUs from a moving FMCW radar. By using a U-Net with a wide spatial receptive field and deep time-frequency feature space, our approach is able to accurately detect and classify moving VRUs against clutter in traffic environments. As a result of the proposed robust pre-processing, our input tensor has a minimal footprint and is invariant to ego-motion. We propose to use objects detected in camera and lidar as weak supervision for training the network parameters. A control network, trained in a fully supervised way using a small set of high quality hand labeled data was used as baseline. We evaluated both models on a content-independent test-set where we observed that the weakly supervised model has a significant edge over the control in terms of average precision. This is not a surprising finding since the weakly supervised model was trained with five times more samples. Even though the weak supervision has an intrinsic uncertainty attached to the labels, we were able to train a more accurate model by incorporating this uncertainty into the loss function. Furthermore, we showed that, regardless of the training regimen, both CNN models outperform classical signal processing algorithms by a wide margin. We suspect that this performance increase comes from the complex CNN model, which was able to extract spatial and micro-Doppler information stemming from VRU motion patterns.

The main weakness of the proposed method is that it relies on supervision from imperfect sensors with known failure modes. It is easily conceivable that training cannot be performed using nighttime recordings since most camera object detectors perform poorly in such circumstances. Moreover, matching camera and lidar detections in the presence of occlusion is ambiguous and results in non-informative training labels. We're currently investigating into better lidar segmentation techniques which would reduce the effect of occlusion by integrating measurements from the past and the future.

The aim of our further research is to improve the performance and robustness of VRU detection by increasing the span and quality of the training dataset. This should be facilitated by firstly capturing sequences in various weather conditions, and secondly by using a higher density lidar and better image object detector for computing training labels. We hope that by making our radar dataset and annotations available[1] we would stimulate the development of better classical and learning based methods for VRU detection.

Acknowledgments

1) This research received funding from the Flemish Government (AI Research Program).
2) The Titan Xp Graphical Card used for this research was donated by the NVIDIA Corporation (Santa Clara, CA, USA) through the Academic Grant Program.

References

[1] J. Dickmann, J. Klappstein, M. Hahn, N. Appenrodt, H. Bloecher, K. Werber, and A. Sailer, ""automotive radar the key technology for autonomous driving: From detection and ranging to environmental understanding"," in *2016 IEEE Radar Conference (RadarConf)*, pp. 1–6, May 2016.

[2] L. Scharf and C. Demeure, *Statistical signal processing : detection, estimation, and time series analysis*. Reading, Mass. Addison-Wesley Pub. Co., 1991.

[3] S. Heuel and H. Rohling, "Pedestrian recognition in automotive radar sensors," in *2013 14th International Radar Symposium (IRS)*, vol. 2, pp. 732–739, June 2013.

[4] O. Schumann, C. Wohler, M. Hahn, and J. Dickmann, "Comparison of random forest and long short-term memory network performances in classification tasks using radar," in *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pp. 1–6, Oct 2017.

[5] X. SHI, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. WOO, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 802–810, Curran Associates, Inc., 2015.

[6] J. Lombacher, K. Laudt, M. Hahn, J. Dickmann, and C. WÃ¶hler, "Semantic radar grids," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1170–1175, June 2017.

[7] D. Brodeski, I. Bilik, and R. Giryes, "Deep radar detector," *CoRR*, vol. abs/1906.12187, 2019.

[8] G. Zhang, H. Li, and F. Wenger, "Object detection and 3d estimation via an FMCW radar using a fully convolutional network," *CoRR*, vol. abs/1902.05394, 2019.

[9] A. Palffy, J. Dong, J. Kooij, and D. Gavrila, "Cnn based road user detection using the 3d radar cube," vol. 5, pp. 1263 – 1270, 01 2020.

[10] B. Major, D. Fontijne, A. Ansari, R. Teja Sukhavasi, R. Gowaikar, M. Hamilton, S. Lee, S. Grzechnik, and S. Subramanian, "Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[11] H. Rohling, "Some radar topics: Waveform design, range cfar and target recognition," 2006.

[12] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous Robots*, vol. 15, pp. 111–127, Sep 2003.

[13] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.

[14] M. Dimitrievski, P. Veelaert, and W. Philips, "Semantically aware multilateral filter for depth upsampling in automotive lidar point clouds," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1058–1063, June 2017.

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015.

[1] Radar data and annotations will become available at: radar-fusion.ipids.ugent.be/ upon publication of the paper.