Multi-sensor System for Driver's Hand-Gesture Recognition

Pavlo Molchanov[1], Shalini Gupta[1], Kihwan Kim[1], and Kari Pulli[2]
[1]NVIDIA Research, Santa Clara, California, USA
[2]Light, Palo Alto, California, USA

***Abstract*— We propose a novel multi-sensor system for accurate and power-efficient dynamic car-driver hand-gesture recognition, using a short-range radar, a color camera, and a depth camera, which together make the system robust against variable lighting conditions. We present a procedure to jointly calibrate the radar and depth sensors. We employ convolutional deep neural networks to fuse data from multiple sensors and to classify the gestures. Our algorithm accurately recognizes 10 different gestures acquired indoors and outdoors in a car during the day and at night. It consumes significantly less power than purely vision-based systems.**

I. INTRODUCTION

In the United States, driver distraction was involved in 22% of the 2.3 million motor vehicle-related injuries and in 16% of the ∼37K fatalities reported in 2008 [1]. Visual-manual interfaces, such as haptic controls and touch screens in cars, cause significant distraction. Hand-gesture-based user interfaces (UIs) in cars can lower visual and cognitive distraction, and can improve safety and comfort. Recent subjective studies suggest that gesture interfaces are desirable to consumers [2]. They can be easily customized to individual users' preferences for gesture types and can be expanded in the future to include functionality for driver monitoring. Work to standardize vehicular gesture interfaces is also underway [3].

Numerous video-based approaches for dynamic gesture recognition have been developed [4], [5], [6]. With the availability of cheap consumer depth cameras, gesture recognition systems using depth cameras have also been introduced [7]. With the exception of a few previous methods [8], [9], most vision-based gesture recognition systems have been developed for environments with controlled illumination [10]. The interior of a car is a challenging environment because the lighting conditions vary a lot. Most consumer color and depth sensors do not work reliably under all these conditions. For example, color sensors are ineffective under low-light conditions at night, while commodity depth cameras that typically use projected IR signals are ineffective under direct bright sunlight. Furthermore, both depth and color sensors suffer from the presence of harsh shadows and hand self-occlusion. Vehicular interfaces also have the added constraint of stricter power efficiency requirements.

Unique micro-Doppler frequency modulation signatures are produced by different types of motion of non-rigid objects [11]. These signatures, as well as the range and the instantaneous angular velocity of the object, can be measured with RAdio Detection And Ranging (*radar*). Compared to color and depth sensors, radars are robust to ambient illumination conditions, have lower cost and computational complexity, and use less power. The radar signal can also penetrate opaque materials such as plastic.

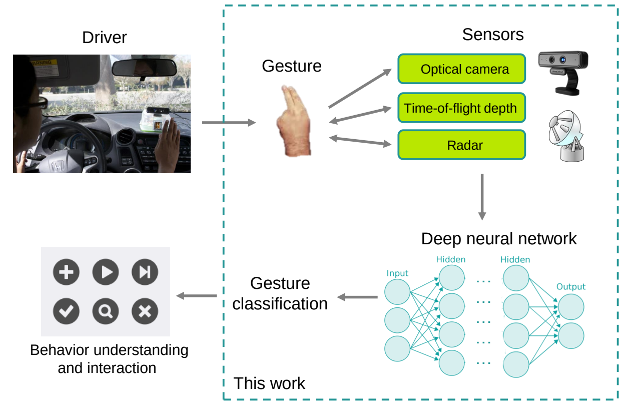

Fig. 1: **Multi-sensor gesture recognition system** We propose a multi-sensor gesture recognition system that uses optical, depth, and radar sensors. Data from the multiple sensors are input into a deep neural network classifier for recognizing dynamic gestures.

Recently convolutional deep neural networks (DNNs) have made a significant impact in computer vision. DNNs have outperformed state-of-the-art machine learning algorithms in very large-scale image recognition [12] and hand-written digits recognition [13]. In a recent competition on multimodal recognition of 20 dynamic gestures from the Italian sign language, an algorithm based on convolutional neural networks [14] ranked first among 17 competing methods [15]. Convolutional DNNs forgo handcrafting discriminatory features for classification, and instead learn them automatically from the training data. DNNs are also attractive for fusing data from multiple sensors because of their ability to automatically weigh their relative importance [16], [17].

We present a novel multi-sensor system comprising of a short-range radar, a color camera, and a time-of-flight (TOF) depth camera for dynamic hand-gesture recognition. Our system detects dynamic gestures with the help of the short-range radar system. Furthermore, it uses a convolutional DNN to fuse data from the three sensors and to classify ten different dynamic hand gestures. An overview of our framework is illustrated in Fig. 1. While imaging sensors [18], [19] or acoustical sensors [4], [7] have been used individually in the past for dynamic hand-gesture recognition, to the best of our knowledge, ours is the first such system to effectively employ all three sensors.

There are various advantages to combining image, depth, and radar sensors. First, it can increase the overall system robustness to varying lighting conditions because it guarantees that data from at least one sensor is reliable under all lighting conditions. Second, since the three sensors provide complementary information about the shape, color, and the instantaneous angular velocity of the hand, they can be combined to improve the classification accuracy of the system. Finally, employing the radar sensor can help to detect and segment dynamic gestures easily and to reduce the power consumption of the system.

In summary, our contributions are: (1) a novel multi-sensor gesture recognition system that effectively combines imaging and radar sensors; (2) use of the radar sensor for dynamic gesture segmentation, recognition, and reduced power consumption; (3) demonstration of a real-time illumination robust gesture interface for the challenging use case of vehicles.

II. Related Work

Video-based hand-gesture recognition algorithms, for numerous applications, have been studied extensively [4], [5]. Recent work also includes depth-based algorithms [7]. Most techniques for dynamic hand-gesture recognition involve temporal localization of the gesture, *e.g.*, by means of a binary “motion” and “no motion” classifier [20]. The hand region in gesture frames is often segmented using color and/or depth information by dense or sparse hand-crafted descriptors [21], and skeletal models are fit to the hand region [10]. To identify the gesture type, sequences of features for dynamic gestures are used to train classifiers, such as Hidden Markov Models (HMM) [6], conditional random fields [22], Support Vector Machines (SVM) [23], or decision forests. Convolutional DNNs have also been employed previously to detect and recognize 20 gestures from the Italian sign language using RGB-D images of hand regions along with upper-body skeletal features [20], and for classifying 6 static hand gestures using depth images [8]. These previous DNN-based gesture recognition methods are different from our proposed work in their data fusion strategies, features employed, and application scenarios.

Most existing approaches for gesture recognition have been developed for controlled lighting conditions where commodity depth and color sensors work well [7]. Gesture recognition becomes challenging in uncontrolled lighting conditions, *e.g.*, in a vehicle, and this problem is much less studied. There exist a few video-based techniques for gesture recognition in cars, that use special IR illuminators and near-IR cameras [24], [25], [2]. In these methods, hand-crafted features, including Hu moments [24], decision rules [25], or contour shape features [2] along with HMM classifiers [25], [24] have been employed. In [9], a system that uses RGBD data, HOG features and an Support Vector Machine (SVM) classifier was proposed. Note that no previous systems for gesture interfaces in cars have employed vision-based and radar sensors together with a DNN classifier.

Independently of vision-based techniques, human motion recognition systems that use micro-Doppler signatures of acoustic signals have also been developed [18], [19], [26]. While acoustical sensors for gesture recognition are not directly applicable inside vehicles because of the presence of significant ambient acoustical noise, their underlying principle of using the unique Doppler signatures for gesture identification has also motivated our work.

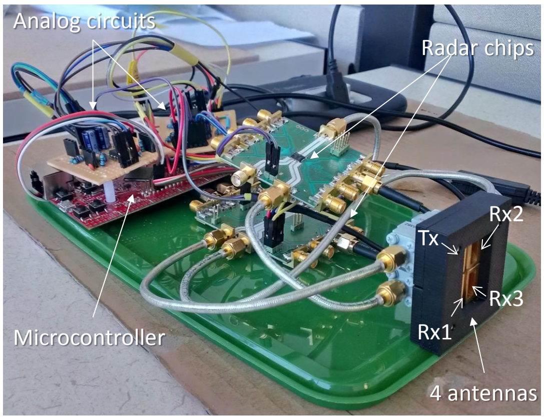

Fig. 2: **The short-range radar prototype**. Our prototype radar module uses a mono-pulse FMCW radar with one transmitting (Tx) and 3 receiving (Rx) antennas. The array of antennas measures the spherical coordinates (distance, azimuth, and elevation) and radial velocity of moving objects.

III. Method

We describe the hardware and algorithmic components of our multi-sensor gesture recognition system.

A. Sensors

Our system uses a color camera, a time-of-flight (TOF) depth camera, and a short-range radar. The color and depth cameras come from the DS325 system (SoftKinetic). The color camera acquires RGB images (640×480) and the depth camera captures range images (320×240) of the objects that are closest to it, both at 30 fps.

Off-the-shelf short-range radar systems in the permitted frequency bands that are appropriate for use inside a car are not widely available. Therefore, we built a prototype radar system, with an operational range of <1m (Fig. 2). The system measures the range (z) and angular velocity (v) of moving objects in the scene, and estimates their azimuth (x) and elevation (y) angles. It employs a mono-pulse Frequency Modulated Continuous Wave (FMCW) signal [27], [28]. The mono-pulse technique allows for the measurement of the angular position of moving objects by employing pairs of vertical (for elevation) and horizontal (for azimuth) co-located receivers. Additionally, the distance of the objects from the radar can be measured by employing the FMCW principle.

In our radar system, we employed a 24GHz front-end Infineon chip and wave guide antennas. This frequency band is available for public use and can be implemented using low-cost (<$40) components. We designed analog circuits for filtering and amplifying the received signal. We used a Tiva C micro controller (Texas Instruments, Dallas, TX) for

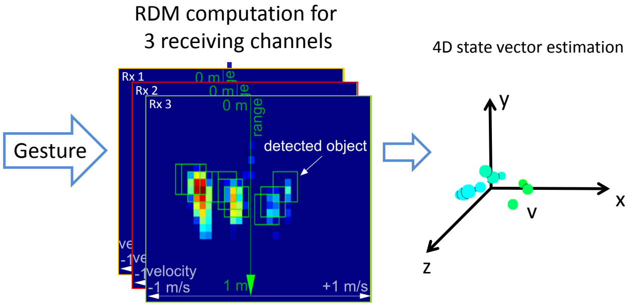

Fig. 3: **The radar signal processing pipeline**. A dynamic gesture generates multiple reflections, which are detected by the range-Doppler maps of all three receivers. By comparing the phases of the signals received by pairs of receivers, a 4D vector comprising of the spatial coordinates and the radial velocity is estimated for each point detected in the RDM.

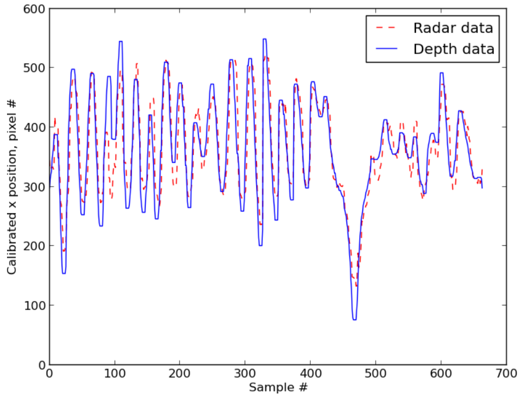

Fig. 4: **Joint calibration of radar and depth sensors.** The plot presents the x coordinates of the center of the calibration target measured by the radar and depth sensors after calibration.

controlling the radar chip, sampling the signal, generating the control signal, and for transferring data to the host. The radar system consumes <1W power from the USB port. It is currently not optimized for power efficiency, and with further design improvements it can consume significantly less power (∼15mW [29]).

The signal-processing pipeline for the radar is illustrated in Fig. 3. A Range-Doppler Map (RDM), which depicts the amplitude distribution of the received signals for certain range (z) and Doppler (v) values, is generated for each of the three receivers. A rigid moving object appears as a single point in the RDM and a non-rigid object, *e.g.*, a hand, appears as multiple points. The radar system can only disambiguate moving objects that are spatially separated in the RDM. Moving objects are detected in the RDM by applying the Cell-Average Constant False Alarm Rate (CA-CFAR) thresholding-based detector [30]. The phase of each detected moving object at the three receivers is compared to estimate its azimuth and elevation angles.

B. Interface

Our gesture interface is located in the central console facing the interior of the car within arm's reach (50cm) of the driver (Fig. 1). It simultaneously captures data of the moving hand with the color camera, the depth camera, and the radar. Since the radar signal can penetrate plastic, it can be housed inside the dashboard. Gestures can be performed anywhere roughly within the center of the field of view (FOV) of the interface.

C. Calibration

We propose a procedure to jointly calibrate the depth and the radar sensor. This calibration procedure is required to register data from multiple sensors, each of which are in their own coordinate system, to a common frame of reference. The calibration is performed only once after the multi-sensor system is installed rigidly.

We assume that a rigid transformation exists between the optical imaging centers of the radar and depth sensors. In order to estimate this transformation, we concurrently observe the 3D coordinates of the center of a moving spherical ball of radius 3cm with both sensors. The best-fit rigid transformation between the 3D coordinates of the ball observed by the two sensors is estimated using the linear least-squares optimization. With the help of the transformation function, the radar data is transformed to the depth camera's coordinate frame. This procedure successfully registers the depth and radar data, as shown in Fig. 4.

D. Gesture Detection

We assume that a true gesture occurs only when the radar detects significant motion, *i.e.*, with velocity above a configurable threshold (0.05m/s), roughly in the center of the FOV of the UI. Since the radar system directly measures the velocity of moving objects, it is the only sensor that is used to detect and segment gestures in our system. The duration of a true gesture is assumed to be between 0.3 and 3 seconds. The gesture ends when no motion is observed by the radar continuously for 0.5 seconds.

In our system, we operate the radar in the always-ON mode and switch ON the color and depth cameras only for the duration of the gesture, *i.e.*, while the radar observes motion. Since the radar consumes significantly less power (<1W) than the optical and depth cameras (<2.5W), our design significantly lowers the overall power requirement of the gesture interface.

E. Feature extraction

We first segment the hand region in the depth image by assuming that it is the closest connected component to the depth camera and generate a mask for the hand region. We normalize the depth values of the detected hand region to the range of $[0, 1]$. We convert the RGB image of the hand obtained from the color sensor to a single grayscale image with values in the range of $[0, 1]$. Note that we did not segment the hand region in the color images.

Using the calibration information between the radar and the depth camera, we register the radar data to the depth

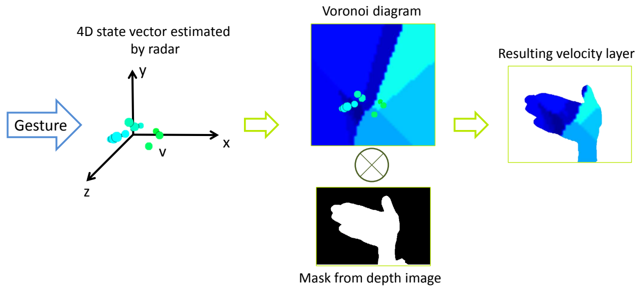

Fig. 5: **Radar feature extraction.** The sparse velocity values of moving objects (left) obtained from the radar are extrapolated across the hand region extracted from a depth camera (middle). The velocity layer (right) is then fed into the DNN.

images. By doing so, we obtain instantaneous angular velocity values for a sparse set of moving objects in the scene (Fig. 5). We then extrapolate these sparse velocity values over the entire FOV of the depth camera, using Voronoi tessellation. We apply the mask for the segmented hand region to the velocity image to retain the velocity values of the hand region only. We call this the radar *image*. Finally, the depth, grayscale and radar images are resized to 32×32 pixels before they are input to the classifier.

We represent a dynamic gesture by a batch of temporal frames, which is input to the classifier for gesture recognition (Fig. 6). For our proposed algorithm, each frame contains three channels, one for each of the three sensors. The classifier requires inputs of constant size, *i.e.*, equal number of frames for each gesture, but in reality the duration of the observed dynamic gestures is variable. Hence, we temporally normalize the gestures to 60 frames by re-sampling them via nearest neighbor interpolation. For example, if the original gesture contains 80 frames, every 4^{th} frame is removed and if the gesture contains 45 frames, every 3^{rd} frame is repeated.

F. Classifier

We train a convolutional deep neural network classifier for recognizing different types of dynamic gestures.

1) Structure: The DNN consists of two 3D convolutional layers, which automatically learn discriminatory spatio-temporal filters to reduce the dimensionality of the input gesture data (Fig. 6). Both convolutional layers contain 25 kernels of size $5 \times 5 \times 5$ and hyperbolic tangent activation functions. Max-pooling layers that retain only the maximum values in blocks of size $2 \times 2 \times 2$ follow each of the convolutional layers. Two fully-connected layers follow the second max-pooling layer. They have linear rectified activation functions and contain 1024 and 128 neurons, respectively.

The output layer implements multi-class logistic regression using a softmax function and produces posterior class-conditional probabilities for each gesture type. The final decision is made by selecting the class with the maximum posterior probability. There are nearly 7.8 million tunable weights in the network that need to be learnt.

2) Initialization: We initialized the weights of the first two 3D convolution layers with random samples from a uniform distribution between $[-W_b, W_b]$, where

$$W_b = \sqrt{\frac{6}{n_i + n_o}}, \tag{1}$$

and n_i and n_o are the number of input and output neurons, respectively. We initialized the biases of the first two layers with 0; the weights of the fully-connected hidden layers with random samples from a normal distribution $\mathcal{N}(0, 0.01)$ and the biases with a value of 1; and the weights and biases of the output softmax layer to 0.

3) Training: We learned the parameters of the DNN by means of a labelled training data set using the Theano package [31]. Training was performed on a CUDA capable Quadro 6000 NVIDIA GPU. DNN training involved the minimization of the negative log-likelihood function via stochastic gradient descent optimization with mini-batches of 20 training samples. The parameters of the network were updated at each back-propagation step i as

$$\lambda_i = \frac{\lambda_0}{1 + i\alpha}, \tag{2a}$$

$$v_i = \underbrace{\mu v_{i-1}}_{\text{momentum}} - \underbrace{\lambda_i \left\langle \frac{\delta E}{\delta w} \right\rangle_{batch}}_{\text{learning}}, \tag{2b}$$

$$w_i = \begin{cases} w_{i-1} + v_i - \underbrace{\gamma \lambda_i w_{i-1}}_{\text{weight decay}} & \text{if } w \text{ is a weight} \\ w_{i-1} + v_i & \text{if } w \text{ is a bias} \end{cases} \tag{2c}$$

where λ_0 is the initial learning rate, μ is the momentum coefficient, $\langle \frac{\delta E}{\delta w} \rangle_{batch}$ is the gradient value of the cost function with respect to the parameter w_i averaged over the mini-batch, and γ is the weight decay parameter. The values of the training parameters (Eq. 2) were selected by cross-validation and were set to $\lambda_0 = 0.01$, $\mu = 0.9$, and $\gamma = 0.0005$.

In order to improve the generalization capability of the network, we trained the DNN with drop-out [32]. During drop-out, the outputs of the second, third, and fourth layers were randomly set to 0 with $p = 0.5$, and subsequently were not used in the back-propagation step of that iteration. For the forward propagation stage, the weights of the layer following the dropped layer were multiplied by 2 to compensate for the effect of drop-out.

We trained the network for 200 epochs. To avoid overfitting we employed early stopping by selecting the network configuration, which resulted in the least error on the validation data set.

We found that a number of procedures helped to increase the accuracy of the system. Weight decay and dropout prevented the network from over-fitting to the training data and improved the classification accuracy by 2.3% on average. Augmenting the training dataset with transformed versions of the training samples also helped to improve the generalization capability of the DNN. We applied the same transformation to all the three sensor channels of each

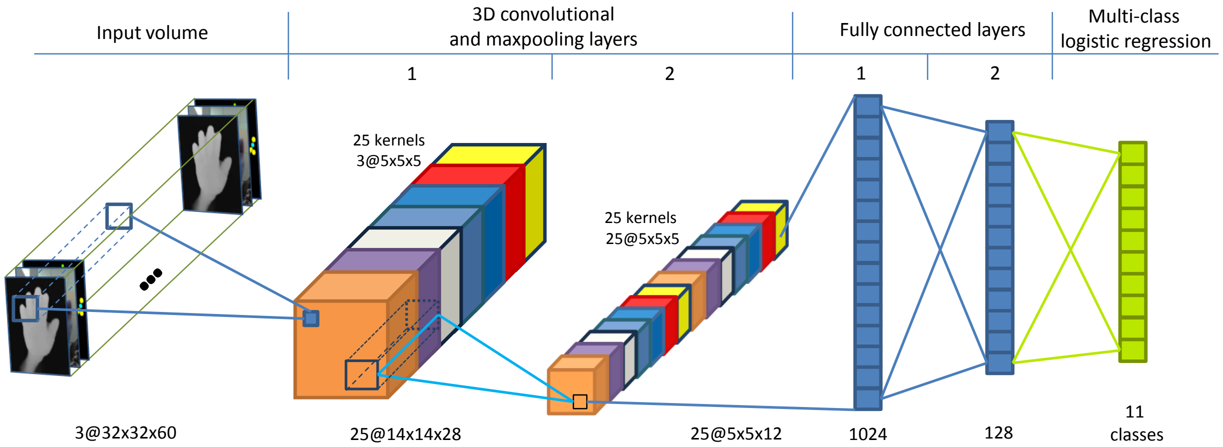

Fig. 6: **DNN pipeline.** The deep neural network classifier used for dynamic gesture recognition. Numbers on the bottom show dimensionality of the data at the output of the corresponding layer (number of channels@XYT)

gesture, which included (a) adding salt and pepper noise, (b) random uniform temporal shifts of ±5 frames or temporal scaling between 80-120% of the entire gesture, or (c) random uniform spatial shifts between ±2 pixels, rotation between ±10 degrees, or scaling between 80-120% of all frames of the gesture.

IV. Evaluation

We describe the experiments that we conducted to evaluate the performance of our hand-gesture recognition system.

A. Database

We collected gesture data indoors in a driving simulator and outdoors in a real car (Fig. 7). For safety reasons, all gestures were performed with the car in the parked position. We acquired data (a) indoors in artificial lighting at night and under indirect sunlight during the day, and (b) outdoors in a car under direct/indirect sunlight during the day and in the absence of light at night (Fig. 8). We collected data in 10 distinct recording sessions. A session included several repetitions of each gesture performed by one subject in a particular environment.

The database contained a total of 1714 gestures of 3 subjects. The gestures included left/right/up/down palm motion, shaking of the hand, CW/CCW hand rotations, left/right swipe, and calling (Fig. 9). Each subject performed 10-20 repetitions of every gesture. In addition to these 10 premeditated gestures, we also acquired a set of random hand motions of the subject.

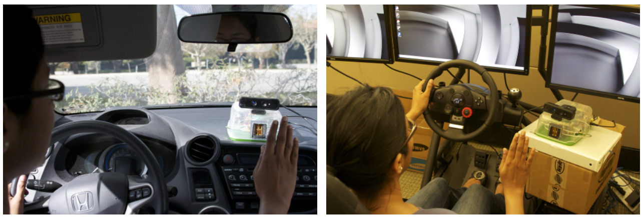

Fig. 7: **Experimental setup.** We used two different experimental setups for gesture data acquisition, *i.e.*, outdoors in a real car (left) and indoors in a driving simulator (right).

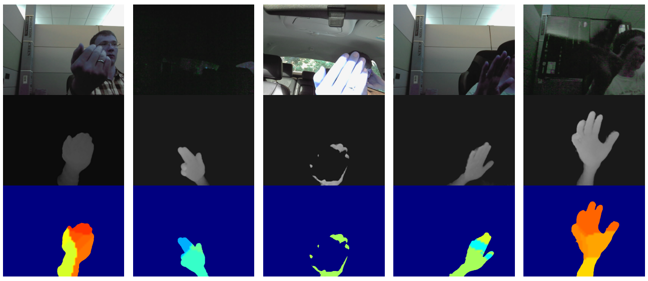

Fig. 8: **Examples of gesture inputs.** Each column represents a different environmental condition: indoors, nighttime inside a car, daytime inside a car, daytime indoors, and nighttime indoors from left to right. The inputs from different sensors are shown in each row: optical, depth, and radar from top to bottom. The colors in the third row indicate the instantness angular velocity measured by the radar sensor.

B. Results

We evaluated the performance of the gesture classification system for two experiments with different partitioning of the database. We performed leave-one-session-out and leave-one-subject-out cross-validation. We evaluated the performance of the DDNs with input from individual sensors, pairs of sensors and all three sensors. When a sensor was not used, its input values were set to zero. We individually trained and tested the DNNs for different sensor types and their combinations.

We computed the average $Precision$, $Recall$, and $Fscore$, and the $accuracy$ of the gesture recognition system. $Precision$ is defined as $TP/(TP + FP)$, where TP and FP are the number of true and false positives, respectively. $Recall$ is defined as $TP/(TP + FN)$, where FN is the number of false negatives. The $Fscore$ is defined as $2 * Precision * Recall/(Precision + Recall)$. We estimated these values for each of the 11 gesture classes and then averaged them together to produce single values. In addition,

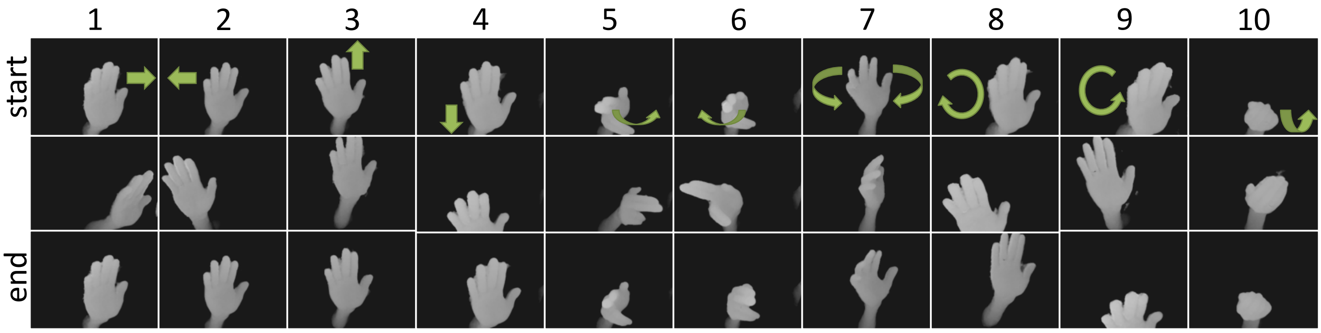

Fig. 9: **Gesture types** We used 10 different dynamic gestures for training our system: moving left/right/up/down (classes 1-4), swiping left/right (classes 5-6), shaking (class 7), CW/CCW rotation (classes 8-9), and calling (class 10).

TABLE I: The classification performance (%) of leave-one-session-out cross-validation for different input sensors.

	O	D	R	DR	DO	RO	DRO
Precision	70.3	92.4	90.0	92.9	93.1	93.3	**94.7**
Recall	60.1	91.5	90.0	91.9	92.4	92.9	**94.2**
Fscore	63.1	91.6	89.3	92.3	92.5	93.0	**94.3**
Accuracy	60.1	90.9	89.1	91.7	92.1	92.6	**94.1**

we calculated the *accuracy* of the system as the proportion of test cases that were correctly classified.

1) Leave-one-session-out cross-validation: In this experiment we left out each of the 10 gesture recording sessions from the training set once. We split the gestures from the left-out session evenly (50/50) into validation and test sets. We averaged, taking into account the number of samples, the results of all sessions to generate the aggregate performance statistics for the system. This experiment was designed to evaluate the generalization performance of the classifier to data acquired under different lighting conditions.

The classification performance of DNNs with different sensors for this experiment is presented in Table I. Among the individual sensors, the best results were achieved by the depth sensor ($accuracy$ = 90.9%), followed by the radar sensor ($accuracy$ = 89.1%). The worst performance was achieved by the optical sensor ($accuracy$ = 60.1%). Employing two sensors improved the accuracy relative to the individual sensors: DR increased the accuracy of the individual depth and radar sensors by 0.7% and 0.8%, respectively, DO by 0.9% and 1.2%, and RO by 3.7% and 2.5%. The best overall performance ($accuracy$ = 94.1%) was achieved by a combination of all three sensors. This network achieved an accuracy of 3.2% higher than the depth only sensor. Lastly, note that the addition of the radar sensor to the depth and optical sensors (DO) improved its accuracy by 2%.

The confusion matrix for the network with all three sensors (DRO) for leave-one-session-out cross-validation is shown in Table II. Observe that most classes were classified correctly. The highest miss-classification rate of 17.7% was observed when class 8 (shake) was miss-classified as class 6 (swipe left).

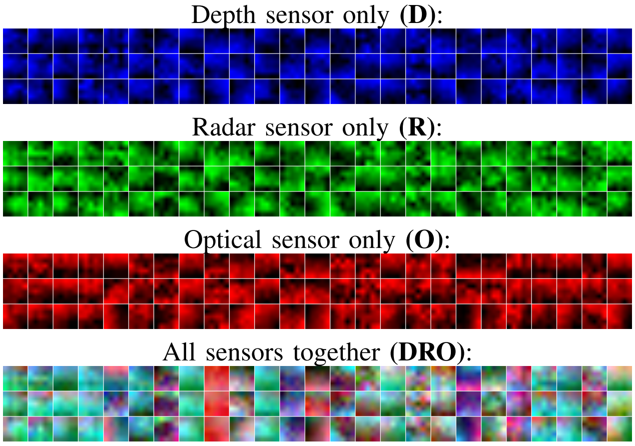

Fig. 10: **Projections of the kernels learned at the first 3D convolution layer.** Each sensor is represented by colors: Depth—blue, Radar—green, Optical—red. Each column is a kernel. The three rows are projections on to the yt, xt, and xy planes, respectively.

The kernels learned by the first convolutional layer of the DRO network are illustrated in Fig. 10. Assuming that x and y are spatial dimensions, and t is the temporal dimension, projections of the learnt convolutional kernels on to the yt, xt, and xy planes are depicted. Observe that all three sensors contributed towards the final decision made by the network. This suggests that the depth, radar, and optical sensors encode complementary information, which helps to improve the accuracy of gesture recognition.

2) Leave-one-subject-out cross-validation: We also evaluated our system's performance for the leave-one-subject-out task with all three sensors contributed to the decision making (DRO network). This experiment helps to evaluate the generalization capability of our system to gestures of unseen subjects. We reserved data from each of the three subjects in our database and trained with data from the two remaining subjects. Our gesture recognition system achieved a classification accuracy of 75.1 ± 5.4% in this experiment.

The confusion matrix of the DRO network for leave-one-subject-out cross-validation is shown in Table III. The lowest correct classification rate was observed for class 11 (call). The up gesture (class 4) was frequently miss-classified as

TABLE II: Confusion matrix of the multi-sensor network (Depth+Radar+Optical) for leave-one-session-out cross-validation.

Truth	Decision										
	Unknown	Left	Right	Up	Down	Swipe Left	Swipe Right	Shake	CW	CCW	Call
Random	**93.3**	0	1.1	2.2	0	0	1.1	0	0	0	2.2
Left	0	**97.8**	2.2	0	0	0	0	0	0	0	0
Right	0	0	**100.**	0	0	0	0	0	0	0	0
Up	10.9	0	0	**89.1**	0	0	0	0	0	0	0
Down	5.9	0	0	0	**94.1**	0	0	0	0	0	0
Swipe L	0	0	0	0	0	**97.1**	1.5	0	0	0	1.5
Swipe R	0	0	0	0	0	0	**97.**	0	0	0	3.
Shake	1.6	0	0	4.8	0	11.3	0	**82.3**	0	0	0
CW	1.8	0	0	0	0	0	0	0	**98.2**	0	0
CCW	0	1.5	0	0	0	0	0	0	1.5	**97.**	0
Call	4.9	1.6	0	0	0	0	3.3	0	0	0	**90.2**

a random gesture (class 1). The shake gesture (class 8) was miss-classified as a swipe left gesture 34.5% of the times. The left palm motion, CW and CCW rotation, and the swipe left gestures were classified correctly most of the time. Observe also that none of the random gestures were miss-classified as a premeditated gesture.

3) Comparison to other methods: Our gesture recognition system is designed to operate inside a car under varied lighting conditions. Ohn-Bar and Trivedi also proposed a solution for this problem using RGBD data [9]. They compared a number of different feature extraction techniques together with a SVM classifier. They obtained their best results with HOG and HOG^2 features extracted from the segmented gesture's video and an SVM classifier with the χ^2 kernel function. We implemented their technique on our data with the following modifications: (a) to fit our dataset we used gestures of size 32×32×60 as inputs to the classifier; (b) instead of RGBD data we used gray-scale and depth images; and (c) we selected the scaling parameter γ for the χ^2 kernel function and the regularization parameter C for training the SVM classifier using a grid search performed on the validation set. For the HOG features, we evaluated cell sizes of 4×4, 8×8 and 16×16 pixels and obtained the best results for cells of size 8×8.

On our dataset, Ohn-Bar and Trivedi's method resulted in accuracies of 88.2% and 51.8%± 21.83% for the leave-one-ssession-out and leave-one-subject-out cross-validation experiments, respectively. For both experiments, our proposed algorithm outperformed their method by 5.9% and 23.3%, respectively.

A comparison of the correct classification rates of various classifiers for gesture sessions conducted under different lighting conditions is presented in Table IV. For our method, observe that adding the optical sensor to the DR network at night did not change the accuracy of the system. For data acquired in the evening and during the day under shadows, the optical sensor improved the accuary by 1.5%. During the day, under bright sunlight, adding the optical sensor considerably improved the accuracy by 13.4%. Ohn-Bar and Trivedi's method shows comparable performance in the evening and during the day under shadows, where all sensors provide reliable data. However, at night where the intensity data is unreliable and during the day under bright sunlight where the depth data is unreliable, the performance of their algorithm decreases. This result suggests that in comparison to SVMs DDNs are more affective at merging partially reliable data from multiple sensors.

TABLE IV: The correct classification rates (%) for the DR and DRO DNN-based classifiers and Ohn-bar and Trivedi's method [9] for sessions recorded under different lighting conditions.

	DR	DRO	[9] (DO)
Night	93.3	**93.3**	77.8
Evening	97.0	**98.5**	97.54
Day (shadow)	90.3	**91.7**	87.0
Day (sunlight)	79.1	**92.5**	79.1

4) Power consumption: An off-the-shelf CUDA implementation of our gesture recognition system ran in 52ms on a Quadro 6000 NVIDIA GPU. Our system requires only the lower-powered (1W) radar to be ON constantly, while the imaging sensors (2.5W) only need to be switched ON for the duration of a gesture. Assuming that 10 gestures/hour are performed for an average duration of 2s each, our design results in ∼50% reduction in power (1.14W) versus an always-ON pure imaging (depth and optical) solution (2.5W). Furthermore, a power-optimized version of the radar prototype (15mW) would result in ∼16x lower power (0.154W) consumption versus a purely imaging system.

V. CONCLUSIONS

In this paper, we introduce a novel multi-sensor system that recognizes dynamic gestures of drivers in a car. Our preliminary experiments demonstrate that the joint use of short-range radar, color, and depth sensors improves the accuracy, robustness, and power consumption of the gesture-recognition system.

In the future, we will explore using the micro-Doppler signatures measured by the radar as features for gesture recognition. We will also expand the study to a larger data set

TABLE III: Confusion matrix of the multi-sensor network (Depth+Radar+Optical) for leave-one-subject-out cross-validation.

	Decision										
Truth	Unknown	Left	Right	Up	Down	Swipe Left	Swipe Right	Shake	CW	CCW	Call
Random	**100**	0	0	0	0	0	0	0	0	0	0
Left	0	**100**	0	0	0	0	0	0	0	0	0
Right	9.0	0	**66.9**	0	0	0	18.0	0	6.1	0.0	0
Up	27.2	0	0	**60.8**	12.0	0	0	0	0	0	0
Down	17.1	0	0	0	**76.3**	0	0	0	6.6	0	0
Swipe L	0	0	0	0	0	**85.9**	0	2	0	0	12.1
Swipe R	0	0	0	0	0	0	**79.2**	0	0	0	20.8
Shake	3.3	0	0	5.0	2.5	34.5	0	**48.9**	2.5	0	3.3
CW	0	0	0	0	0	0	0	0	**95.6**	10	4.3
CCW	0	2.8	0	0	8.5	0	0	0	0	**88.6**	0
Call	19.3	0	0	6.4	5.0	29.9	19.9	0	0	0	**19.3**

of gestures of more subjects to improve the generalization of the DNN, and develop methodologies for continuous online frame-wise gesture recognition.

REFERENCES

[1] NHTSA, "Traffic safety facts," National Highway Traffic Safety Administration, Tech. Rep., 2009.

[2] F. Parada-Loira, E. Gonzalez-Agulla, and J. Alba-Castro, "Hand gestures to control infotainment equipment in cars," in *IEEE Intelligent Vehicles Symposium*, 2014, pp. 1–6.

[3] A. Riener, A. Ferscha, F. Bachmair, P. Hagmüller, A. Lemme, D. Muttenthaler, D. Pühringer, H. Rogner, A. Tappe, and F. Weger, "Standardization of the in-car gesture interaction space," in *International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. ACM, 2013, pp. 14–21.

[4] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Systems, Man, and Cybernetics*, vol. 37, no. 3, pp. 311–324, 2007.

[5] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *PAMI*, vol. 19, pp. 677–695, 1997.

[6] T. Starner, A. Pentland, and J. Weaver, "Real-time american sign language recognition using desk and wearable computer based video," *PAMI*, vol. 20, no. 12, pp. 1371–1375, 1998.

[7] J. Suarez and R. R. Murphy, "Hand gesture recognition with depth images: A review," in *RO-MAN*. IEEE, 2012, pp. 411–417.

[8] K. R. Konda, A. Königs, H. Schulz, and D. Schulz, "Real time interaction with mobile robots using hand gestures," in *ACM/IEEE HRI*, 2012, pp. 177–178.

[9] E. Ohn-Bar and M. Trivedi, "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations," *Intelligent Transportation Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–10, 2014.

[10] J. J. LaViola Jr., "An introduction to 3D gestural interfaces," in *SIGGRAPH 2014 Courses*, 2014.

[11] C. Clemente, A. Balleri, K. Woodbridge, and J. Soraghan, "Developments in target micro-Doppler signatures analysis: radar imaging, ultrasound and through-the-wall radar," *EURASIP Journal on Advances in Signal Processing*, no. 1, 2013.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.

[13] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural Comput.*, vol. 22, no. 12, pp. 3207–3220, 2010.

[14] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, "Multi-scale deep learning for gesture detection and localization," in *ECCV ChaLearn Workshop on Looking at People*, 2014.

[15] S. Escalera, X. Bar, J. Gonzlez, M. A. Bautista, M. Madadi, M. Reyes, V. Ponce, H. J. Escalante, J. Shotton, and I. Guyon, "Chalearn looking at people challenge 2014: Dataset and results," in *ECCV ChaLearn Workshop on Looking at People*, 2014.

[16] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 689–696.

[17] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," in *Advances in neural information processing systems*, 2012, pp. 2222–2230.

[18] S. Gupta, D. Morris, S. Patel, and D. Tan, "Soundwave: using the Doppler effect to sense gestures," in *CHI*, 2012, pp. 1911–1914.

[19] K. Kalgaonkar and B. Raj, "One-handed gesture recognition using ultrasonic Doppler sonar," in *ICASSP*, 2009, pp. 1889–1892.

[20] N. Neverova, C. Wolf, G. Paci, G. Sommavilla, G. W. Taylor, and F. Nebout, "A multi-scale approach to gesture detection and recognition," in *Computer Vision Workshops (ICCVW)*, 2013, pp. 484–491.

[21] P. Trindade, J. Lobo, and J. Barreto, "Hand gesture recognition using color and depth images enhanced with hand angular pose data," in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*, 2012, pp. 71–76.

[22] S. B. Wang, A. Quattoni, L. Morency, D. Demirdjian, and T. Darrell, "Hidden conditional random fields for gesture recognition," in *CVPR*, vol. 2, 2006, pp. 1521–1527.

[23] N. Dardas and N. D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 11, pp. 3592–3607, 2011.

[24] M. Zobl, R. Nieschulz, M. Geiger, M. Lang, and G. Rigoll, "Gesture components for natural interaction with in-car devices," in *Gesture-Based Communication in Human-Computer Interaction*. Springer, 2004, pp. 448–459.

[25] F. Althoff, R. Lindl, L. Walchshausl, and S. Hoch, "Robust multimodal hand-and head gesture recognition for controlling automotive infotainment systems," *VDI BERICHTE*, vol. 1919, p. 187, 2005.

[26] S. Dura-Bernal, G. Garreau, C. Andreou, A. Andreou, J. Georgiou, T. Wennekers, and S. Denham, "Human action categorization using ultrasound micro-doppler signatures," in *Human Behavior Understanding*. Springer, 2011, pp. 18–28.

[27] J. Woll, "Monopulse Doppler radar for vehicle applications," in *Intelligent Vehicles Symposium*, 1995, pp. 42 –47.

[28] I. Lozano Mrmol, "Monopulse range-doppler FMCW radar signal processing for spatial localization of moving targets," Universidad Politcnica de Cartagena, Tech. Rep., 2012.

[29] Infineon, "Using BGT24MTR11 in low power applications, 24 GHz Radar," Application Note AN341, Tech. Rep., 2013.

[30] H. Rohling, "Radar CFAR thresholding in clutter and multiple target situations," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. AES-19, no. 4, pp. 608–621, 1983.

[31] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Python for Scientific Computing Conference (SciPy)*, 2010.

[32] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv*, 2012.