Chapter 18
Reliability-Aware and Robust Multi-sensor Fusion Toward Ego-Lane Estimation Using Artificial Neural Networks

Tran Tuan Nguyen, Jan-Ole Perschewski, Fabian Engel, Jonas Kruesemann, Jonas Sitzmann, Jens Spehr, Sebastian Zug, and Rudolf Kruse

Abstract In the field of road estimation, incorporating multiple sensors is essential to achieve a robust performance. However, the reliability of each sensor changes due to environmental conditions. Thus, we propose a reliability-aware fusion concept, which takes into account the sensor reliabilities. By that, the reliabilities are estimated explicitly or implicitly by classification algorithms, which are trained with extracted information from the sensors and their past performance compared to ground truth data. During the fusion, these estimated reliabilities are then exploited to avoid the impact of unreliable sensors. In order to prove our concept, we apply our fusion approach to a redundant sensor setup for intelligent vehicles containing three-camera systems, several lidars, and radar sensors. Since artificial neural networks (ANN) have produced great results for many applications, we explore two ways of incorporating them into our fusion concept. On the one hand, we use ANN as classifiers to explicitly estimate the sensors' reliabilities. On the other hand, we utilize ANN to directly predict the ego-lane from sensor information, where the reliabilities are implicitly learned. By the evaluation with real-world recording data, the direct ANN approach leads to satisfactory road estimation.

Keywords Information fusion · Reliability · Neural networks · Ego-lane estimation · Intelligent vehicles

T.T. Nguyen (✉) · J.-O. Perschewski · F. Engel · J. Kruesemann · J. Sitzmann · J. Spehr
Volkswagen Aktiengesellschaft, Wolfsburg, Germany
e-mail: Tran.Tuan.Nguyen@volkswagen.de; Jan-Ole.Perschewski@volkswagen.de; Fabian.Engel@volkswagen.de; Jonas.Kruesemann@volkswagen.de; Jonas.Sitzmann@volkswagen.de; Jens.Spehr@volkswagen.de

S. Zug · R. Kruse
Faculty of Computer Science, Otto-von-Guericke University Magdeburg, Magdeburg, Germany
e-mail: Sebastian.Zug@ovgu.de; Rudolf.Kruse@ovgu.de

É. Bossé, G. L. Rogova (eds.), *Information Quality in Information Fusion and Decision Making*, Information Fusion and Data Science,
https://doi.org/10.1007/978-3-030-03643-0_18

18.1 Introduction

Advanced driver assistance systems (ADAS) and automated driving heavily rely on environment perception and especially on road estimation. By that, current research explores various algorithms toward road detection by using multiple sensors, such as camera, radar, lidar, etc. Thereby, one of the biggest challenges is the huge variety of environmental conditions that influence sensor performances. This leads to sensor failures in several scenarios. For instance, a camera-based detection system can provide sufficient results under many weather conditions. However, this system can fail in case of heavy rain, snow, etc. In contrast, radar sensors can detect the surrounding objects despite these conditions since their technology is not affected by rain or snow as cameras. For that reason, it is necessary to combine the data of distinct sensors so that the system can constantly produce sufficient results.

In our previous works, we introduce a multi-source fusion framework for robust ego-lane detection [1–3]. Thereby, we take into account that the sensor reliabilities depend on environmental conditions and can change over time. The reliabilities are estimated by applying different classification algorithms, which are offline trained by using the extracted information from sensors' detections. Consequently, the fusion process based on *Dempster-Shafer theory* incorporates these reliabilities to combine the information of the sources.

In this work, we exploit the possibility of estimating the ego-lane directly by using neural networks. By that, the reliabilities are internally learned by the networks and encoded as weights of the neurons. This differs from the approaches of Nguyen et al. in [3, 4], where the reliability of each source is estimated by training a separate classifier. Furthermore, we integrate new environment information to take advantage of the redundant sensor system, such as detections from a surround view camera system, free space information, etc. To achieve higher accuracy of the classification, we utilize the mutual information of the features to select the features with the greatest influence on the classification. Finally, we evaluate our presented approaches by using a new database of real-world data recordings.

This work is organized as follows: Sect. 18.2 explains three categories of perception approaches toward automated driving and gives an overview of various works. In Sect. 18.3, we introduce our concept of incorporating reliabilities into ego-lane estimation by using different classifiers. Following, Sect. 18.4 applies neural network to explicitly learn the sensors' reliabilities. Afterward, Sect. 18.5 explains our approach of using neural networks to directly estimate the ego-lane. Lastly, Sect. 18.6 presents the experimental results obtained for the feature selection, the reliability estimation, and the final ego-lane estimation.

18.2 Related Work

The approaches in the field of automated driving can be divided into three categories [5], illustrated in Fig. 18.1. The first category consists of *behavior reflex approaches*, which use purely *data-driven techniques*, also called as *AI techniques*, to map sensor data to driving decisions directly. The second category with *direct perception approaches* apply AI algorithms to estimate a selected set of features representing the relevant information of the current environment. Afterward, a simple controller uses these features to realize driving functions. Representing the third category, *mediated perception approaches* build an environment model by processing the sensor data using both *model-based* methods and AI techniques, respectively. Based on the generated environment model, AI methods are utilized to derive the driving actions of the vehicle. Following, all categories will be discussed in detail.

18.2.1 Behavior Reflex Approaches

In the early stages of automated driving, Pomerleau et al. propose a behavior reflex approach using an *artificial neural network* (ANN) to estimate the steering angle for an intelligent vehicle [6]. Thereby, the network consisting of only three layers is trained by using a low-resolution 30 × 32 pixel camera image. Thus, the input layer of the ANN contains 960 neurons. Following, the input layer is fully connected to the hidden layer consisting of five neurons, which in turn is connected to the output layer of 30 neurons. Each neuron in the last layer represents a steering angle that is used to calculate the steering of the vehicle. To provide a stable behavior, the final steering angle is determined by calculating the center of masses of the activations around the highest activated neuron.

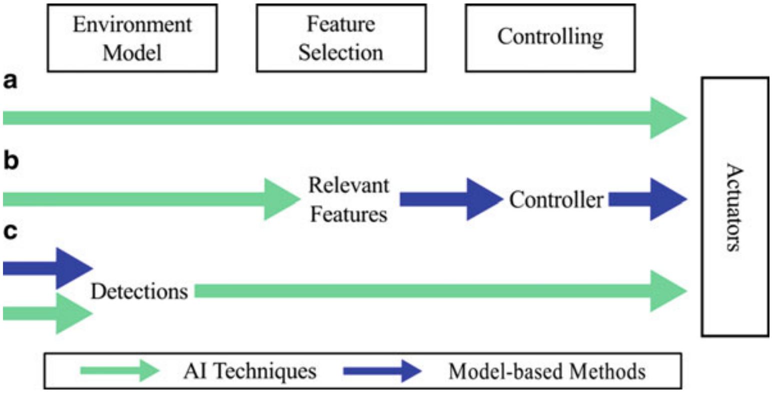


Fig. 18.1 Perception models: (**a**) Behavior reflex, (**b**) Direct perception, and (**c**) Mediated perception approaches

A more sophisticated approach using ANNs is presented by Bojarski et al. [7]. Their system uses *convolutional neural network* (CNN) as recent advances of ANNs. For that, they use the images of three cameras to determine the steering wheel angle. In order to train the multilayer CNN, backpropagation is performed with the mean squared error of the estimated angle to the angle chosen by a human driver. Additionally, they rotate and shift the images to avoid an overfitting to the training data. In their evaluation, they reach an autonomy level of 98%, which is defined as follows:

$$\text{autonomy Level} = \left(1 - \frac{\text{\#interventions} \cdot 6 \text{ seconds}}{\text{elapsedTime}}\right) \cdot 100 \tag{18.1}$$

A similar approach using CNN to determine the steering angle is presented by Chen et al. [8]. The resulting network is able to perform steering with a mean error of 2.42°. However, the authors explain that evaluating the camera images frame by frame is not appropriate since the repetition of the small error in every frame can result in leaving the lane. Thus, they conclude that it is necessary to incorporate temporal information into the network to improve the results in a continuing scenario.

Codevilla et al. [9] propose a more practice-oriented approach by incorporating commands into the learning process. Therefore, they use a camera system which determines the steering angle and acceleration using a CNN. Furthermore, they compare two architectures for their networks. On the one hand, the command input architecture combines the image processing results, the measurements of the environment, and the command by feeding the outputs into fully connected layers, which determine the action. On the other hand, the branched architecture combines the image processing results and environment measurements and forwards the outputs into fully connected layers depending on the command. Impressively, the branched version drove an off-the-shelf 0.20 scale truck nearly perfectly on walkways in a residential area.

The problems of using behavior reflex approaches are that it is hardly possible to install a fail-safe. This can result in accidents in unknown environments and endanger other traffic participants.

18.2.2 Direct Perception Approaches

In [5], Chen et al. introduce a direct perception approach for autonomous driving by choosing a set of 13 features to represent the current environment. These features contain information about the angle between the vehicle and the road, distances to lane markings, and preceding vehicles on other lanes. Using these features, the authors construct a controller, which minimizes the distance to the lane center line and keeps a safe distance to other traffic participants. In order to determine the features, they use two different approaches: a handcrafted GIST system [10] and

CNN. A as result, CNN outperforms the GIST system regarding every parameter. Using the superior CNN, they develop a system that can perform well in both virtual and real environments. Although this approach seems to achieve good results, two problems can occur. First, the controller depends strongly on correct inputs, which cannot be ensured in the current state. Secondly, if this approach needs to be scaled to fully autonomous driving, the selected features will become as complex as in mediated perception approaches. Therefore, the simple controllers will not be sufficient and should be replaced by mediated perception approaches.

Similar to [5], Al-Qizwini et al. provide a different direct perception approach called GlAD [11]. Therefore, they compare the top three CNN architectures, namely, GoogLeNet [12], VGGNet [13], and Clarifai [14]. These CNNs are used to learn five affordance parameters, which are used by the controller to drive the intelligent vehicle. During the training of the CNNs on images provided by TORCS, GoogLeNet outperforms VGGNet and Clarifai. Hence, they use GoogLeNet as the best network to evaluate the automated driving capability in a simulated environment by measuring the mean and deviation to the lane center. Their algorithm performs well and achieves a mean deviation on the evaluation tracks of at most 0.2 m. Although this approach seems to be promising, it suffers from the lack of complexity in comparison to real-world scenarios because of using simulation results. By way of example, they cannot simulate all mistakes that other traffic participants could make to react accordingly.

18.2.3 Mediated Perception Approaches

Mediated perception approaches are characterized by modeling a complex environment representation when combining information from several sensors. Thereby, the biggest challenge is how to handle inconsistency and conflict between the information coming from different sources. Thus, several works investigate the sensor reliability by using different methods, e.g., classifiers [3, 15, 16] and failure models [17]. At the decision layer, these reliabilities can be exploited to fuse only reliable sources.

Frigui et al. present a context-dependent multisensor fusion framework [18]. By that, they use a clustering algorithm to cluster the extracted features. Each cluster represents a certain context and contains data that shows similar characteristics of the environment. Afterward, a reliability of each source is manually defined for each context. This approach can be problematic when the number of features rises, and the clustering algorithms will suffer from the curse of dimensionality. In this case, the number of clusters would rise exponentially.

In [15], Hartmann et al. fuse multiple sensors to create a road model, which is then verified with a digital map. Therefore, they train an ANN using a large database containing sensor data and the associated map geometry. The goal is to assess whether the estimated road model is incorrect and does not match with the digital map. This can be the case when the predicted road course changes due to construction works or errors of the detection algorithms. As a result, the trained ANN outputs a reliability value representing the probability for an error between the estimated road and the digital map. This approach can detect contradictions, but it cannot decide which source is faulty [19]. Hence, this method could be improved by identifying the incorrect source [20].

Realpe et al. introduce a fault-tolerant object estimation framework [21]. First, objects are separately estimated by using data from each single sensor. For each sensor, the discrepancy of its estimated objects to the reference in the offline evaluation phase is used as weight for the final fusion. This concept is promising, but the reliability estimation could be further increased by using additional context information, such as the road type, where the vehicle is driving on.

Romero et al. present an environment-aware fusion approach for lane estimation [22]. By that, they compare the estimated lane from each sensor with the ground truth. Based on the comparison result, they assign a reliability value to each sensor for the current GPS position. When the vehicle is located at a certain position, the stored reliabilities are used to perform a weighted fusion. However, this approach is not generalizable to new areas since it uses GPS position to predict reliabilities and requires the vehicle to have been there before. Instead of utilizing GPS coordinates, additional features extracted from sensor detections could be used to make the estimations location-independent [19].

The discussed works in this chapter contain interesting approaches, but they still have potential for improvement or are quite work-intensive. Hence, the following chapter will explain our fusion concept.

18.3 Overall Concept

Our fusion concept is an extension of our previous work in [3, 23]. As illustrated in Fig. 18.2, it consists of multiple levels such as in the JDL model [24]. At *Level 0*, the raw sensor data is preprocessed on the basis of physical signal level. At *Level 1*, multiple detection modules iteratively utilize the preprocessed data to estimate and predict the states of different object types. This includes tasks such as object detection, tracking, association, etc. The low-level fusion, e.g., object association of different sensors [25, 26], is taking place here. In our work, the used sensors are delivered with their internal processing modules and provide different results such as lane markings, dynamic objects, etc.

Starting from *Level 2*, we present two different fusion concepts, where reliable sources should be preferred over unreliable sources. In the first approach represented in Sect. 18.4, we utilize artificial neural networks (ANNs) to estimate the reliability of different ego-lane models by using the scenario features, which are extracted from the sensor and contextual information. Afterward, the fusion based on Dempster-Shafer theory utilizes these estimated reliabilities to identify and neglect the unreliable sources. In the second approach, we utilize ANNs to directly estimate the

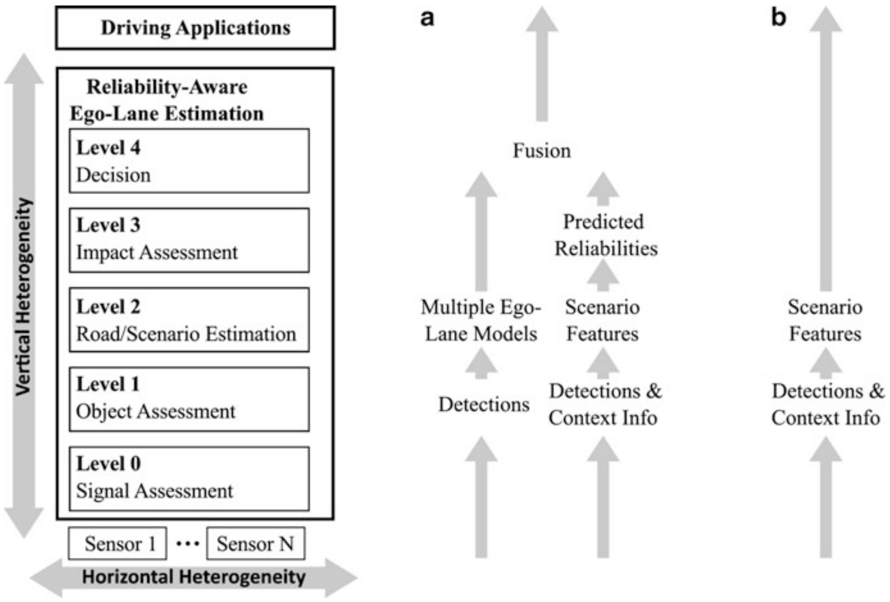


Fig. 18.2 Overview of our two different fusion concepts. While (**a**) estimates the reliabilities of the separately estimated ego-lane models and incorporates them into the fusion, (**b**) estimates the ego-line directly using sensor detections

ego-lane (Sect. 18.5). By that, the network should internally learn the reliabilities of the sources for an optimal estimation. Both concepts are detailed in their respective sections. Since the scenario features are used by both approaches, we will explain them in the following.

18.3.1 Sensor Setup

As shown in Fig. 18.3, we use a setup of three-camera systems in order to detect lane markings. Thereby, each camera system separately provides estimations for the next right lane marking (*RM*) and the next left lane marking (*LM*). In this work, a prefix of "*second*" or "*third*" denotes the affiliation to that particular camera system. If no prefix is given, the estimation belongs to the first camera system. Furthermore, the prototype vehicle also is also equipped with several radar and lidar sensors for a 360° object detection, which is not be further explained here.

By way of example, Fig. 18.4 shows four scenarios with the detected lane markings and objects. The highway in Fig. 18.4a demonstrates an ideal scenario, where all lane markings can be perceived clearly. Thereby, the two front-facing camera systems can detect markings up to 100 m, while the third camera system has a shorter detection range of about 20 m. In this scenario, the vehicle can use

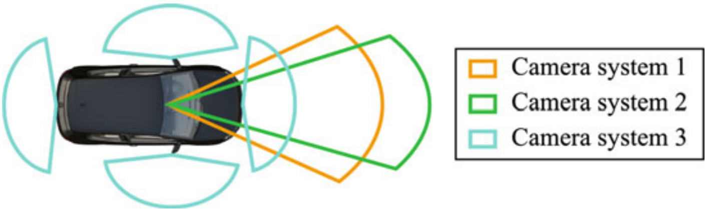


Fig. 18.3 The prototype vehicle with three-camera systems: Two are front-facing and differ slightly in field of view; the third consists of four fish-eye cameras for a surround view. The positions of other sensors such as lidars, radars, and ultrasonic sensors are not shown here

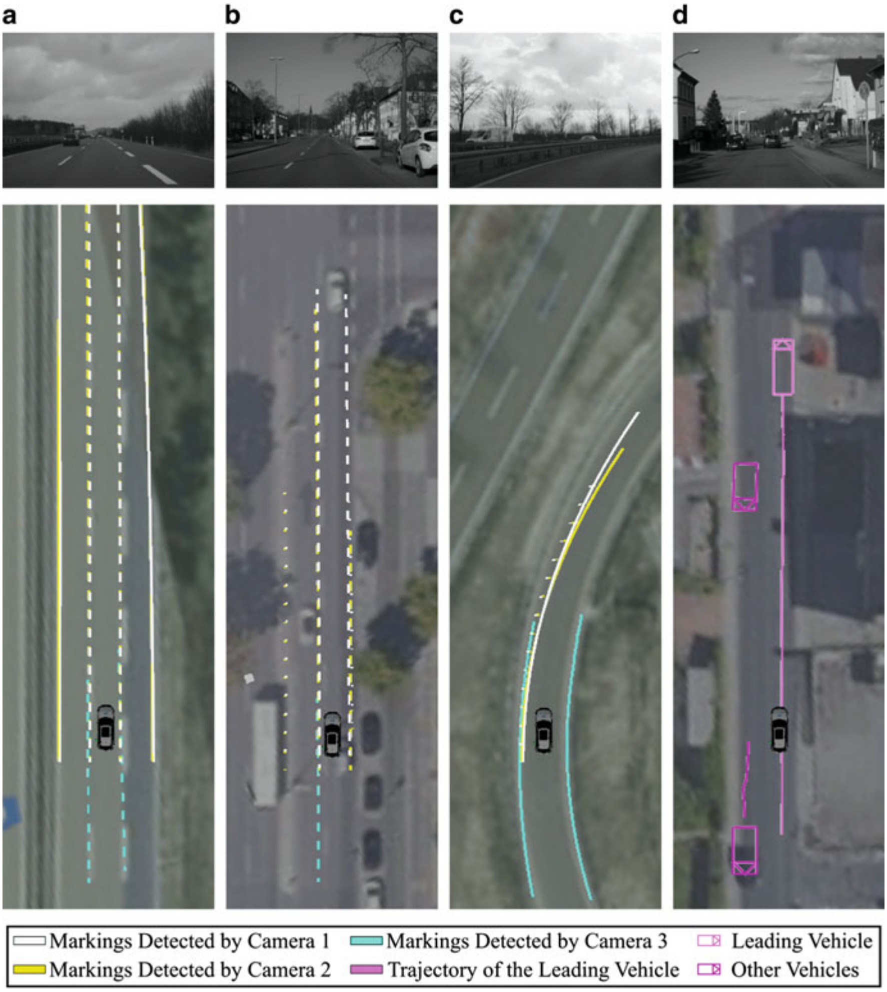

a
b
c
d
Markings Detected by Camera 1
Markings Detected by Camera 2
Markings Detected by Camera 3
Trajectory of the Leading Vehicle
Leading Vehicle
Other Vehicles

Fig. 18.4 First row: images from the first camera. Second row: visualization of detection results of all three cameras and object estimations on *Google Maps*

any marking from the first two cameras or a combination of them to estimate the current ego-lane. As opposed to this, Fig. 18.4b depicts an urban scenario, where the detection ranges of all cameras are smaller than in the highway scenario. Moreover, markings are not existing on the right side so that only Camera 1 and Camera 2 can identify the curbstone as lane boundary. In contrast, the left lane marking is perceived clearly by all cameras. Therefore, the vehicle should orientate to the left lane marking. Especially in the on-ramp scenario in Fig. 18.4c, the third camera system outperforms the rest by detecting markings on both sides up to 20 m away. Here, the first two camera systems cannot recognize the right marking due to their narrow field of views (Fig. 18.4). In order to handle this scenario, the vehicle should utilize the detected markings of Camera 3. Last, Fig. 18.4d depicts another urban scenario with no markings on both sides. Unfortunately, none of the cameras can detect the curbstone stably. Only the leading vehicle can be detected so that its trajectory should be used to generate an ego-lane hypothesis.

18.3.2 Scenario Features

This section explains in detail the composition of the scenario features, which we extract from sensor and context information. In this work, all lane markings as well as the trajectory of the leading vehicle (*ACC object*) are modeled by an approximation of the clothoid model [27]:

$$y(x) \approx \phi_0 \cdot x + \frac{C_0}{2}x^2 + \frac{C_1}{6}x^3 \tag{18.2}$$

$$= a_1 \cdot x + a_2x^2 + a_3x^3 \tag{18.3}$$

A subset of the used scenario features is generated from these clothoid parameters, which can be seen in Table 18.1. Additionally, this table contains a likelihood ξ, representing a measure of uncertainty about the existence of an object. Moreover, Table 18.1 also contains the estimated lane width $Lane_w$, the feature *free*, that expresses the amount of free space along the clothoid evaluated with an occupancy grid built by using lidar data. Furthermore, we introduce several consensus features

Table 18.1 Sensor-related and consensus features of all markings and the trajectory of the leading vehicle: $h \in \{LM, RM, SLM, SRM, TLM, TRM, ACC\}$

Feature	Description	Feature	Description
$h\ x_0$	Start on the x-axis	$h\ y_0$	Lateral offset
$h\ l$	Clothoid length	$h\ \phi$	Clothoid angle
$h\ c_0$	Curvature	$h\ c_1$	Curvature change
$h\ \xi$	Existence likelihood	$Lane_w$	Lane width
$CS_h\ l$	Deviation to l	$CS_h\ \phi$	Deviation to ϕ
$CS_h\ c_0$	Deviation to c_0	$CS_h\ c_1$	Deviation to c_1

Table 18.2 Motion parameters of object $o \in \{Ego, ACC\}$

Feature	Description	Feature	Description
$o\ x$	x-position	$o\ v_x$	Longitudinal velocity
$o\ y$	y-position	$o\ v_y$	Lateral velocity
$o\ v_\phi$	Yaw rate	$o\ v_{LM}$	Velocity to *LM*
$o\ turn$	Tri-state	$o\ v_{RM}$	Velocity to *RM*

measuring the deviations to respective average values. Last, the type of all lane markings is also utilized, e.g., solid, dashed, and curbstone.

For moving objects like the ego-vehicle and the leading vehicle, we extract various motion parameters, as seen in Table 18.2.

Furthermore, we utilize external contextual features extracted from a navigation map. These include *roadType* (e.g., highway, rural, urban, connection), *linkType* (e.g., ramp, roundabout), *laneClass* (e.g., normal, split, merge, intersection), and *cityLimitStatus* (e.g., inside, outside). Additional features are the mean width $\mu_{EgoLaneWidth}$ and the standard deviation $\sigma_{EgoLaneWidth}$ of the ego-lane.

Instead of using these features directly to train ANNs as in [3], we normalize these features and encode them to reach a higher classification performance, which is described in the following section.

18.3.3 Preprocessing Features

If a sensor provides data directly to ANN, the input can suffer from artificial semantic through different ranges and meanings of the data. For example, comparing the $roadType$ that are denoted by natural numbers, the distances between two categories are varying even though the semantics are not different. Thereby, the difference between a highway and an urban scenario is equal to the difference between a highway and a rural scenario. Therefore, the distance between these categories should not differ. For that reason, we apply *one-hot encoding* to the categorical input data. By that, a one-hot encoding transforms a categorical feature with n categories into a vector of n entries, where each entry is set to one if the index corresponds to the respective category and to zero otherwise as

$$\text{one} - \text{hot} : \{0, 1, \ldots, n-1\} \to \{0, 1\}^n \ , \ \text{one} - \text{hot}(k)_i = \begin{cases} 1 & i = k \\ 0 & \text{else} \end{cases} \tag{18.4}$$

Another challenge is the huge variety of ranges in the data set. For instance, the length l of the lane markings can reach up to 100 m, while the angle ϕ varies between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$. Hence, l has a bigger influence on the results until the network learns to reduce its influence by adapting the weights. Therefore, the convergence of the network is slower than the case where alldata is in similar ranges. For that reason, we apply the following *min-max-scaling* to each feature, so that all values are in the interval $[-1, 1]$ with:

$$\text{scale}(x) = \frac{2 \cdot (x - \min_x)}{\max_x - \min_x} - 1 \tag{18.5}$$

18.4 Reliability Estimation

This section presents the application of ANNs as reliability estimators to the reliability-aware road fusion framework of Nguyen et al. [3, 23]. For this purpose, this section starts by explaining the fusion framework and the model-based ego-lane generation in greater detail. Following, we select for each ego-lane model the most important features, which are obtained by applying the feature selection method mutual information (MI). Afterward, we present the structure and training process of ANNs based on chosen features and introduce different fusion strategies.

18.4.1 Concept

Sections 18.2 and 18.3 clarify the relevance of fusing multiple sources for road estimation. By that, a proper incorporation of reliabilities can leverage the fusion's performance [19]. Thus, we present a multisensor fusion framework, which continuously estimates the sensor reliabilities and uses them to perform the fusion. Adapted from [23], Fig. 18.5 shows different layers of the framework, whereby the contributions of this work are highlighted in green.

At *Layer 0*, different sensor inputs are processed. This preprocessed data is then passed to *Layer 1*, where different types of information are estimated, e.g., lane markings, free space information,vehicles, etc.

At *Level 2*, several hypotheses for the current ego-lane are generated using a model-based approach from Toepfer et al. [1]. Additionally, here we also generate the *scenario features*, which are extracted from sensor detections and contextual information. By way of example, the parameters describing the lane markings are selected, such as the length, the curvature, etc. Moreover, we extend the feature set from [28] with the *consensus features*, which describe the similarity among the lane markings and the driven trajectory of the leading vehicle.

In the offline phase of the *Level 3*, the estimated ego-lane hypotheses are compared with the ground truth, which is represented by the driven trajectory of human drivers. If the deviation from the ground truth exceeds a predefined threshold, the hypothesis will be considered as unreliable and vice versa. Together with the corresponding features, they are stored in a database to train different classifiers. By that, one classifier is trained to predict the reliability of each ego-lane model. During the online phase, each estimated ego-lane is assigned with a predicted reliability from the corresponding classifier.

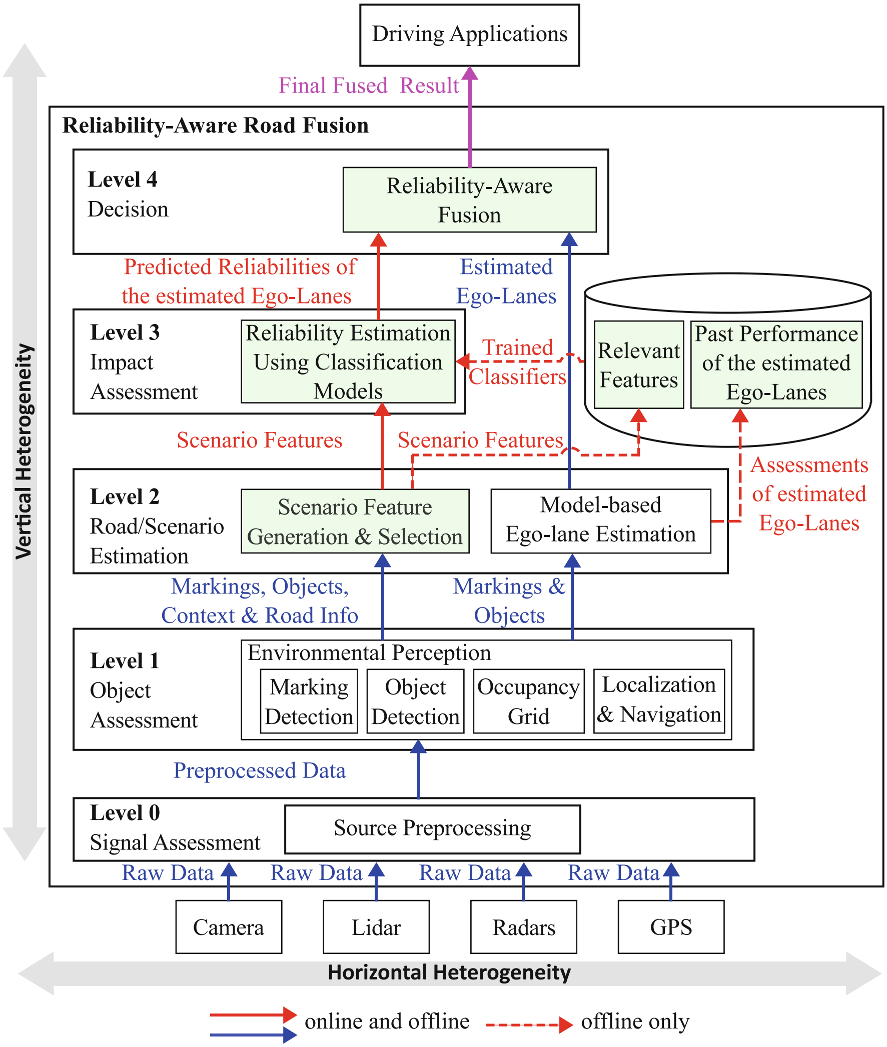


Fig. 18.5 Reliability estimation and reliability-aware fusion as an additional supervision system within the road estimation task [23] (Blue: Data for road detection; Red: Reliability information)

As the last layer, *Level 4* fuses different models depending on the predicted reliabilities. Following, the final ego-lane estimation is then used to perform driving functions.

In this work, we apply *mutual information (MI)* to detect nonlinear relations between the scenario features and the reliability values [29]. Additionally, ANNs are employed as reliability estimators since they perform well in many other tasks and could increase the reliability estimation result [5, 30, 31].

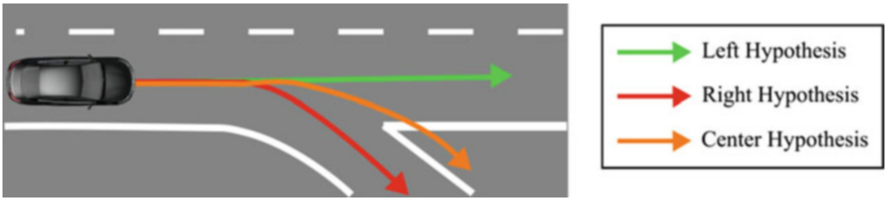


Fig. 18.6 Three estimated ego-lane hypotheses for each camera system [2]

18.4.2 Hypotheses

This section introduces different types of ego-lanes, which are created from lane markings and leading vehicles. Thereby, the detection of lane markings is performed independently for each camera system. In general, the results of each system are used to generate three model-based ego-lane hypotheses (Fig. 18.6). By that, the *left hypothesis (LH)* and the *right hypothesis (RH)* use only the left and right lane markings, respectively. The *center hypothesis (CH)* utilizes the detected lane markings on both sides. By applying this process to the three-camera systems, we can receive up to nine ego-lane estimations. Additionally, the *vehicle hypothesis (VH)* represents the trajectory of the leading vehicle as shown in Fig. 18.4d. This leads to the following set H of hypotheses, where the prefixes "*F*," "*S*," and "*T*" indicate the first, second, and third camera system, respectively:

$$H = \{FLH, FRH, FCH, SLH, SRH, SCH, TLH, TRH, TCH, VH\}$$

18.4.3 Feature Selection

Since information from multiple sources is incorporated, the generated feature vector consists of hundreds of elements. Training classifiers with all these features would be computationally expensive, and the results can worsen due to the curse of dimensionality [32]. Moreover, not all features directly affect the reliabilities. Therefore, we perform a feature selection so that only the most relevant features are used to train the classifiers.

For this work, we apply the method *mutual information (MI)*, which is a measure of the dependency between two variables [29]. It is used to determine the information about a variable through another variable. For this purpose, MI is not using the covariance like the linear correlation coefficient but the distance between two probability distributions. Hence, MI can describe nonlinear relationships between two variables. Assuming an independent, identical distribution of a set of N bivariate measurements $\{t_i = (x_i, y_i) \mid i = 1, \ldots, N\}$ of the features $X = \{x_1, \ldots, x_N\}$ and $Y = \{y_1, \ldots, y_N\}$, the mutual information of X and Y is defined as follows:

$$I(X,Y) = \int\int p(x,y)\ \log \frac{p(x,y)}{p(x)p(y)} dxdy \tag{18.6}$$

where $p(x, y)$ is the joint probability density and $p(x)$ and $p(y)$ are the marginal probability densities of X and Y, respectively.

Since the densities are not always known, an approach of approximating MI is applied. Therefore, the values of X and Y are sorted into containers of finite sizes, which is described in the following:

$$I_{\text{cont}}(X,Y) = \sum_{ij} p(i,j)\ \log \frac{p(i,j)}{p(i)p(j)} \tag{18.7}$$

where $p(i, j) = \int_i \int_j p(x, y)\ dxdy$, $p(i) = \int_i p(x)\ dx$, and $p(j) = \int_j p(y)\ dy$. By that, $\int_i$ denotes the integral over container i and $\int_j$ denotes the integral over container j.

The number of entries of each container is counted and

$$p(i) \approx n_x(i)/N \tag{18.8}$$

$$p(i) \approx n_y(i)/N \tag{18.9}$$

$$p(i,j) \approx n(i,j)/N \tag{18.10}$$

are approximated, where $n_x(i)$ and $n_y(i)$ represent the number of entries of the respective container i of X and Y, and $n(i, j)$ denotes the number of overlapping entries. When the number of containers is increased toward to infinity and the size of the containers is aiming toward zero, I_{cont} converges to I.

18.4.4 Training Process

During the offline training phase, the database is divided into training and testing datasets (Fig. 18.7). Afterward, the data is resampled to balance the number of negative and positive samples. As a result, the resampled datesets contain the same amount of samples for both classes to avoid bias during the training [33]. Following, a feature vector $\mathbf{X}_h$ with four different categories for each sample of h is generated as

$$\mathbf{X}_h = [s_h, \tau, \gamma_{\text{int}}, \gamma_{\text{ext}}] \tag{18.11}$$

where s_h describes the sensor information, τ represents the consensus features, and γ_{int} and γ_{ext} denote the internal information (e.g., odometry data) and environment information (e.g., the road type), respectively [23]. After creating $\mathbf{X}_h$, an error metric is applied to the ego-lane hypothesis h to determine the label L_h. By that, L_h will

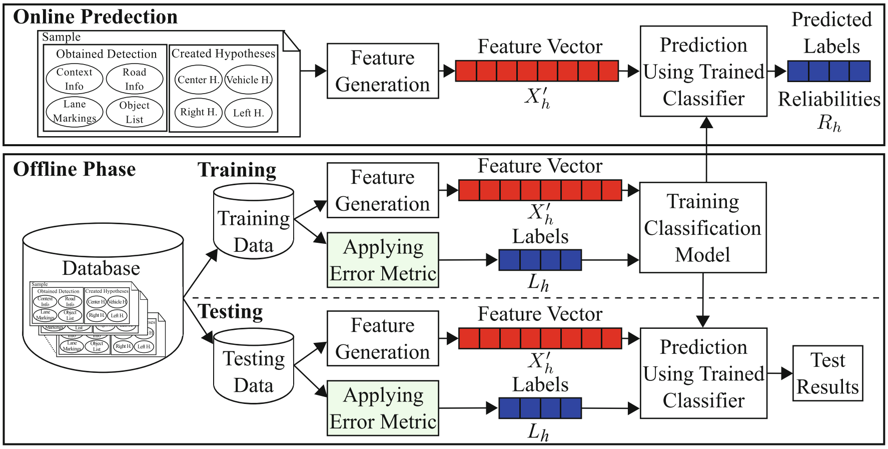


Fig. 18.7 Overview of the application of the classifier [23]

be considered as *reliable* if the deviation of h from the reference is smaller than a predefined threshold. We will explain the used metric in Sect. 18.6.1.

To evaluate the trained networks, the testing dataset is used. Thereby, the created feature vectors are passed directly to the networks and their predictions are compared with the actual test target. The evaluation process and the results are explained in greater detail in Sect. 18.6.

18.4.5 Artificial Neural Networks for Reliability Estimation

In order to estimate the reliability of each hypothesis $h \in H$, we train an ANN ANN_h for each h separately. Thereby, the output of ANN_h represents the estimated reliability R_h. The structure of each individual ANN is shown in Fig. 18.8.

After applying MI to the feature vector $\mathbf{X}_h$, the 25 most relevant features $\mathbf{X}'_h$ are used as the input for the training. Thereby, these features are then preprocessed by the normalization and one-hot encoding described in Sect. 18.3.3. As a consequence, the processed feature vector can have i elements with $i \geq 25$ due to the one-hot encoding. Since the networks are fully connected, each neuron's input function receives the output from all neurons of the preceding layer.

The next five layers consist only of *rectified linear units (ReLU)*, i.e., they employ $f(x) = \max(x, 0)$ as their activation function. These layers only differ by the number of neurons. Starting with 25 neurons in the first layer, the number is reduced by five for every succeeding layer. The last layer has only one neuron and a sigmoid activation function to produce an output between zero and one, which represents the final reliability value of the corresponding ego-lane model.

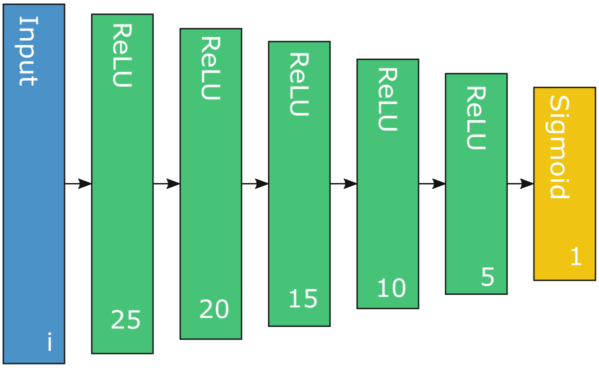


Fig. 18.8 Structure of ANNs toward reliability estimation

During the training, the label vector L_h is compared with the estimation produced by the network to update the weights of the neurons. Since the basic backpropagation algorithm often suffers from contrary training examples and requires a high number of iterations until convergences [32], we apply *Stochastic Gradient Descent (SGD)*, an advanced backpropagation algorithm, to update the weights [34]. Instead of minimizing the total error as the basic backpropagation, SGD minimizes the empirical risk over the training data $D = \{(x_i, y_i) \mid i = 1, \ldots, n\}$ as

$$E(f) = \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i) \tag{18.12}$$

where l denotes the loss function describing the loss of the prediction $f(x_i)$ regarding the target y_i. In this work, we use the squared Euclidean loss function, which is defined as

$$E_{L2}(f) = \frac{1}{2n}\sum_{i=1}^{n} ||f(x_i) - y_i||_2^2 \tag{18.13}$$

For convenience, the loss is divided by two for an easier derivative of the squared Euclidean loss. For an optimal gradient in the learning phase, the gradient has to be calculated in every iteration, which produces a heavy computational effort. Hence, SDG estimates the gradient by using a batch $B \subset D$, which is significantly smaller than D with $|B| \ll |D|$.

$$E_{L2}(f) = \frac{1}{2|B|}\sum_{i=1}^{|B|} ||f(x_i) - y_i||_2^2 \tag{18.14}$$

By that, Eq. 18.14 describes the risk that is optimized in each iteration. During the weight adaption, the learning rate needs to be decreased to achieve convergence. Although better results can be achieved, the gradient descent can still get stuck in a local minimum of the empirical risk [35]. Therefore, a momentum term is used in the weight adaption, which helps the network to converge faster and leave local minima [36]:

$$\Delta w_{t+1} = \mu \Delta w_t - \alpha \nabla E_{L2}(f) \tag{18.15}$$

where w_t and w_{t+1} are the weights, μ is the momentum, Δw_t is the weight change in step t, and α is the base learning rate. This technique can increase the performance of ANNs as described in [37].

To train the networks, we set the base learning rate $\alpha = 0.1$. Every 100,000 iterations, the learning rate α is multiplied with a factor $\gamma = 0.8$ to support the converging of the networks. In total, each network is trained with 1,000,000 iterations using a batch size of $|B| = 4$. The momentum of the weight change is chosen as $\mu = 0.1$.

18.4.6 Incorporating Reliabilities into Fusion

During the online prediction phase, a feature vector is generated for each ego-lane hypothesis. The trained ANNs take these vectors as input and predict the reliability values, which are used to combine the ego-lanes. Thereby, the quality of the fusion is restricted by the quality of the source [16]. However, different fusion strategies create results with varying quality. Therefore, this section presents several basic strategies and a more complex strategy based on Dempster-Shafer theory [3, 20].

18.4.6.1 Basic Strategies

Following, we introduce several basic fusion strategies:

Baseline (BE) The standard road estimation approach from [1] serves as a baseline strategy.

Average fusion (AVG) By this strategy, every estimation model is equally involved in the fusion. This is one of the easiest approaches, but *AVG* will not produce the best results because inferior models can impair the fused result.

Weight-based fusion (WBF) As an extension of *AVG*, the reliability R_h of every model h can be utilized as weight for the fusion. Using R_h allows to disregard unreliable models and focus on the combination of the remaining reliable models.

Winner-take-all (WTA) WTA selects solely the ego-lane model with the greatest R_h, and all other hypotheses are discarded.

Minimum (MIN) The ego-lane model with the smallest R_h, i.e., the most unreliable one, is chosen for the fusion. This strategy is needed to prove that the unreliable sources can be identified by the classifiers and assigned with lower reliabilities.

Random (RAN) As an additional baseline, *RAN* chooses a hypothesis arbitrarily.

18.4.6.2 Dempster-Shafer Theory (DST)

The theory of belief functions was developed by Dempster and Shafer in [38]. Its application is the combination of several unreliable sources to a total result which often occurs in reality. As introduced by Nguyen et al. [3], the reliability of each ego-lane hypothesis h can be modeled as a *frame of discernment* $\Theta_h = \{\rho_h, \bar{\rho_h}\}$, which consists of two statements *Reliable* ρ_h and *Unreliable* $\bar{\rho_h}$. The following steps are taken under the assumption that the reliabilities of the ego-lane models are independent [19]. Since DST also models a belief function for the situation, where both states of Θ_h can occur, adding ρ_h and $\bar{\rho_h}$ does not have to result in one as compared to the classical probability theory. This is difficult to represent using the Bayesian probabilistic model. As a consequence, the power set Φ_h for a hypothesis h is defined as:

$$\Phi_h = 2^{\Theta_h} = \{\emptyset, \{\rho_h\}, \{\bar{\rho_h}\}, \{\rho_h, \bar{\rho_h}\}\} \tag{18.16}$$

where $\{\rho_h, \bar{\rho_h}\}$ describes the occurrence of both possibilities. The mass function for the reliability of the model h at time t is defined as follows:

$$\begin{aligned} &\sum_{\theta \in \Phi} m^t(\theta) = 1 \text{ with } m^t(\emptyset) = 0, \quad m^t(\{\rho_h\}) = R_h^t \cdot PR_h \\ &m^t(\{\bar{\rho_h}\}) = (1 - R_h^t) \cdot PR_h \qquad m^t(\{\rho_h, \bar{\rho_h}\}) = 1 - PR_h \end{aligned} \tag{18.17}$$

where PR_h represents the precision of the neural network ANN_h, which estimates the reliability R_h of h. By that, PR_h is determined by evaluating the classifier ANN_h offline using test data. Assuming two different times t and t' are both independent, the fusion of m^t and m^{t+1} is defined as:

$$m_F(z) = m^t \otimes m^{t+1}(z) = \frac{\sum_{x,y \subseteq \Phi, x \cap y = z} m^t(x) \cdot m^{t+1}(y)}{1 - \sum_{x,y \subseteq \Phi, x \cap y = \emptyset} m^t(x) \cdot m^{t+1}(y)} \tag{18.18}$$

Every hypothesis' reliability consists of two parts. The *belief* b_F and the *plausibility* pl_F. The first describes the belief in the correctness of the hypothesis and the second the plausibility of the hypothesis:

$$b_F(\{\rho_h\}) = \sum_{X \subseteq \{\rho_h\}, X \neq \emptyset} m_F(X) = m_F(\{\rho_h\}) \tag{18.19}$$

$$pl_F(\{\rho_h\}) = \sum_{\rho_h \in X} m_F(X) = m_F(\{\rho_h\}) + m_F(\{\rho_h, \bar{\rho_h}\}) \tag{18.20}$$

To compare the estimated R_h of each hypothesis, the average of *belief* and *plausibility* is used, like in [23, 39]:

$$p_F(\{\rho_h\}) = \frac{b_F(\{\rho_h\}) + pl_F(\{\rho_h\})}{2} \tag{18.21}$$

Using $p_F(\{\rho_h\})$ as the weight for the respective hypothesis and a predefined threshold ϵ_R, only the most reliable hypotheses are allowed to take part in the fusion.

Instead of an explicit reliability estimation, the next section will describe another fusion approach, which estimates the ego-lane directly by using sensors detections.

18.5 Ego-Lane Estimation Using Artificial Neural Networks

18.5.1 Concept

An alternative approach for ego-lane estimation can be performed with artificial neural networks, whose architecture is shown in Fig. 18.9. Hereby, *Level 0*, *Level 1*, and *Level 2* are analogous to the reliability estimation process in Sect. 18.4. By using the generated scenario features from *Level 2*, we apply ANNs as *regressors* to estimate the clothoid parameters of the ego-lane at *Level 3* and *Level 4*. Thereby, we create the training data by taking the human-driven path as a reference, which Sect. 18.5.2 will explain in detail. Moreover, we will present the network structure in Sect. 18.5.3 and the training procedure in Sect. 18.5.4.

18.5.2 Ground Truth Acquisition

An important task is creating the reference data, which is used as targets to train ANNs. For that reason, we use real-world data recordings provided by the test vehicle to determine the necessary coefficients. During a local simulation of the recordings, the positions and the orientations of the vehicle are saved in a database by reconstructing the human-driven path (Fig. 18.10). For time t, the reference is created by an approximation of a clothoid using the points $p_0, \ldots, p_n$. By that, p_0 represents the current vehicle position at time t and $p_1, \ldots, p_n$ the vehicle positions at time $t + 1, \ldots, t + n$. Therefore, the consecutive points $p_1, \ldots, p_n$ are rotated and translated in the coordinate system of p_0. As a result, the reference ego-lane is represented by an approximation of a clothoid [27]:

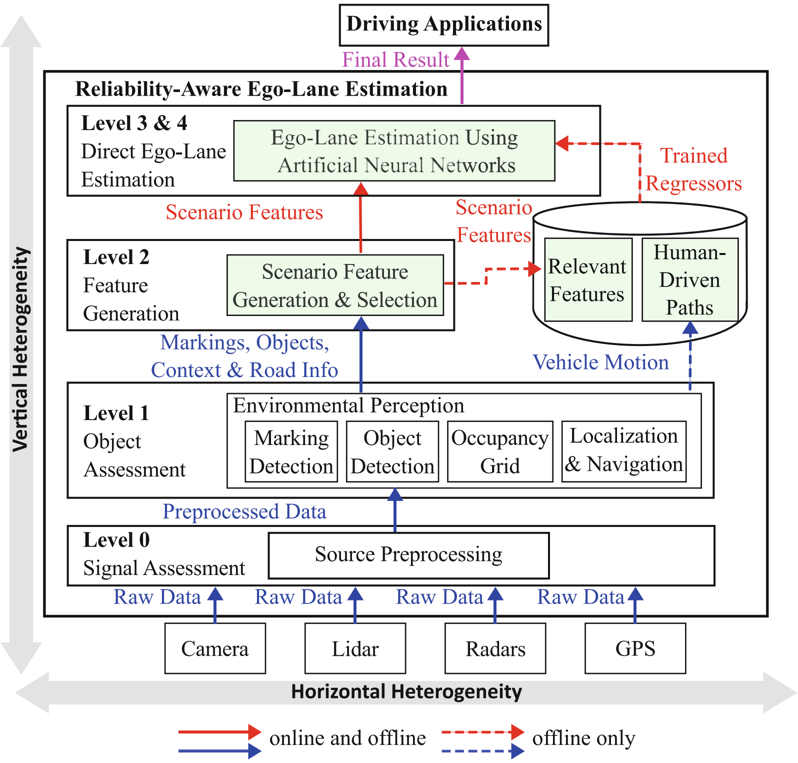


Fig. 18.9 Direct ego-lane estimation using artificial neural networks (Blue: sensor information; Red: reliability information)

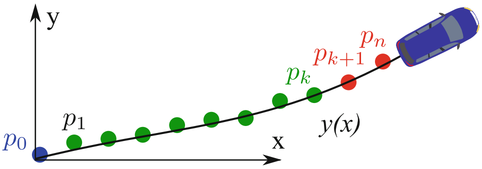


Fig. 18.10 The ground truth at p_0 is acquired by using linear polynomial regression of $p_0, \ldots, p_k$. The points $p_{k+1}, \ldots, p_n$ are available, but they are left out due to exceeding the maximal distance or angle

$$y(x) \approx \phi_0 \cdot x + \frac{C_0}{2}x^2 + \frac{C_1}{6}x^3 \tag{18.22}$$

$$= a_1 \cdot x + a_2x^2 + a_3x^3 \tag{18.23}$$

We determine a_1, a_2 and a_3 by applying linear polynomial regression. Therefore, we construct the following linear system using the consecutive points.

$$\mathbf{y} = \begin{bmatrix} y_{p_1} \\ \vdots \\ y_{p_n} \end{bmatrix} = \begin{bmatrix} x_{p_1} & x_{p_1}^2 & x_{p_1}^3 \\ \vdots & \vdots & \vdots \\ x_{p_n} & x_{p_n}^2 & x_{p_n}^3 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \mathbf{X} \cdot \mathbf{a} \tag{18.24}$$

Next, this system is solved by using Moore-Penrose inverse regression [40] since **X** is most of the time not invertible. For that reason, the parameters can be calculated by using

$$\mathbf{a} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{18.25}$$

Consequently, the coefficients are $\phi_0 = a_1$, $C_0 = 2a_2$, and $C_1 = 6a_3$. Basically, this process could be applied to all consecutive points $p_1, \cdots, p_n$ in the recording, but this is neither representative for an estimation nor applicable due to the computational effort. Hence, we reduce the number of points by choosing a subset of only k consecutive points as

$$\left\{ p_k \mid \sum_{i=1}^{k} \text{distance}(p_{i-1}, p_i) < 50 \wedge |\text{direction}(p_k)| < 15^\circ \right\} \tag{18.26}$$

First, only the first k points that are less than 50 m away from the start point p_0 are selected. Secondly, the orientation of these points has to be smaller than 15° to achieve a sufficient approximation by the polynomial.

Since the manually-driven path is used to calculate the targets to train ANNs, we have to remove samples/situations where the driver leaves the current ego-lane. For example, such samples can be obtained by intersections, lane change, overtaking maneuvers, etc. Additionally, the samples that do not contain any information about the road course are also removed since ANNs cannot produce any useful estimation in such scenarios.

18.5.3 Structure

An important decision is the choice of a structure for ANNs. We also decide to use one network for each parameter to preserve expressiveness. Each ANN has seven layers consisting of a decreasing amount of neurons as displayed in Fig. 18.11. The first layer contains 80, the second 60, the third 40, the fourth 20, the fifth 10, the sixth 5, and the last 1 neuron. All layers except the last layer consist of *rectified linear units (ReLU)*. The last layer has the identity function as activation function to enable an output of arbitrary real numbers. We choose this structure since the layers using ReLU can deal with not linearly separable data. Additionally, the layers decrease in the number of neurons to generalize the scenario features in small steps.

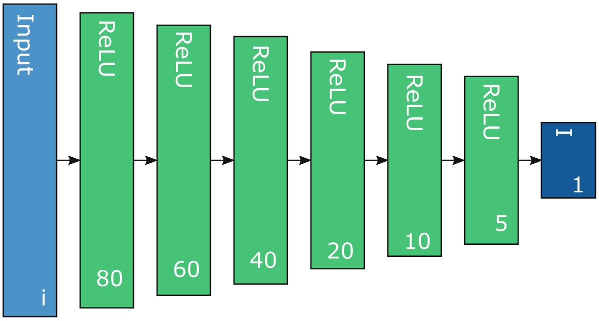


Fig. 18.11 Structure of the ANN to estimate each single parameter of the clothoid model. The text describes the activation function in the layer, where ReLU denotes a layer of rectified linear units and I is the identity function

18.5.4 Training

Analogously to the reliability estimation, we use the stochastic gradient descent from Sect. 18.3 with an Euclidean loss for 100,000 iterations. Using this technique, a learning rate of $\alpha = 0.0001$ that decreases to 0.9 times itself every 10,000 iterations is chosen. Additionally, we chose a momentum weight of $\mu = 0.0000001$ that is multiplied by 0.01 after the same amount of iterations. Moreover, we use a batch size of $|B| = 25$.

During this process, we scale the learning targets by multiplying them by 10000. Hereby, the real appearing value range becomes bigger, so that the impact of the gradient is bigger and leads to faster convergence. Furthermore, the training data set is resampled regarding the $roadType$, so that the trained networks can perform well in each category.

18.6 Experimental Results

In this section, we use real-world data recordings to evaluate our introduced fusion concepts. Figure 18.12 shows the routes, where the prototype vehicle drove in Wolfsburg and its surroundings. Thereby, we planned our routes in order to archive a balanced distribution of highway, ramp, rural and urban scenarios.

First, we will present the evaluation concept. Following, the impact of the feature selection with *mutual information* is analyzed. Afterward, the reliability estimation and the final performance of both fusion concepts are presented.

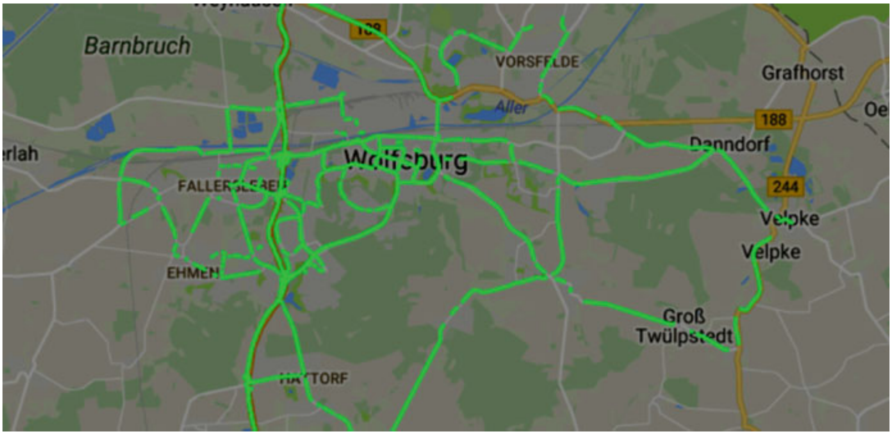
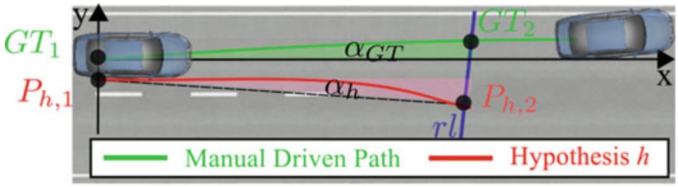


Fig. 18.12 Driven roads for recording training and testing data



$$\boxed{\Delta\alpha = |\alpha_{GT} - \alpha_h|}$$

Fig. 18.13 Metric to measure reliability of the estimated ego-lanes [28]

18.6.1 Concept

In the following, we use the angle metric presented by Nguyen et al. [28] to assess the reliability of the estimated ego-lanes. Instead of using highly-precise DGPS and digital map as the authors in [1, 15, 26], this metric incorporates the human-driven path as reference, which can be reconstructed with standard and cheap motion sensors. As shown in Fig. 18.13, this metric measures the angle deviation $\Delta\alpha$ between the estimated lane and the manually driven path for a run length rl starting from the position of the ego-vehicle at time t. The motivation for this metric is because human drivers cannot drive perfectly on the lane centerline during recording data. This leads always to small lateral offsets between the estimation and the reference, even when the estimation could be detected perfectly [2]. By using the angle deviation $\Delta\alpha$, only the parallelism between the hypotheses and the driven path is taken into account.

As Fig. 18.13 shows, the angle α_h of the hypothesis h is calculated using the position $P_{h,2} = (x_{h,2}, y_{h,2})$ at the run length rl and its start position $P_{h,1} = (x_{h,1}, y_{h,1})$. The ground truth is reconstructed by using the human-driven path, where $GT_1 = (0, 0)$ represents the ego-vehicle's position at time t and $GT_2 = (x_{GT,2}, y_{GT,2})$ denotes the position at time t' with $t' \gg t$. In other words, GT_2 represents the position of the vehicle after driving rl meters. As a result, the angle difference can be calculated as

$$\Delta\alpha = \left| \arctan\left(\frac{y_{h,2} - y_{h,1}}{x_{h,2} - x_{h,1}}\right) - \arctan\left(\frac{y_{GT,2}}{x_{GT,2}}\right) \right| \tag{18.27}$$

By that, we consider an ego-lane estimation as *reliable* if its angle deviation is smaller than 2° for $rl = 30\,\text{m}$. For the sake of completeness, we also use the lateral offset $\Delta d = |y_{GT,2} - y_{h,2}|$ as another criterium when evaluating the hypotheses to be comparable with related works.

18.6.2 Result of Feature Selection

This section discusses the results of feature selection with *mutual information* (*MI*) by showing the 10 highest ranked features for each hypothesis in Figs. 18.14

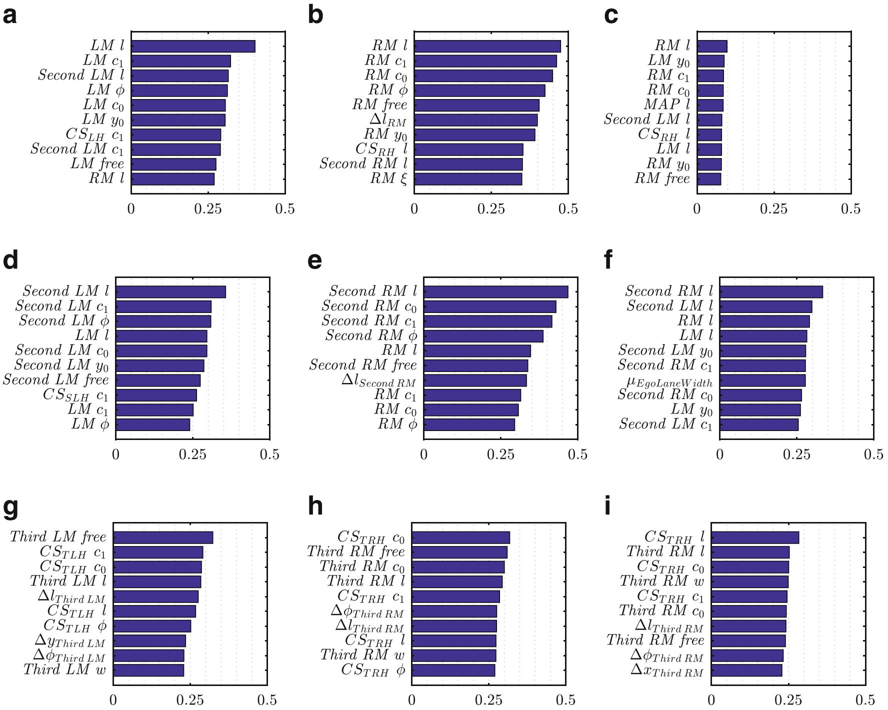


Fig. 18.14 The 10 highest ranked features for ego-lane models, which are generated by using lane markings. (**a**) LH. (**b**) RH. (**c**) CH. (**d**) SLH. (**e**) SRH. (**f**) SCH. (**g**) TLH. (**h**) TRH. (**i**) TCH

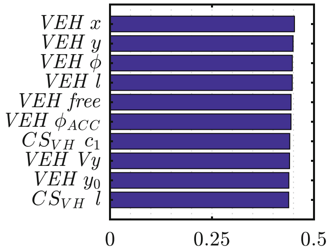


Fig. 18.15 The 10 highest ranked features for *VH*

and 18.15. Thereby, *LM* denotes the left ego-lane marking and *RM* the right ego-lane marking respectively. The prefix *Second* and *Third* represents the camera system, where the lane marking is coming from. Moreover, *VEH* denotes features of the *ACC* object.

As shown in Fig. 18.14, the length l, the curvature c_0, the curvature change c_1, and the yaw angle ϕ of the lane markings are very important for the ego-lane models, which are created by involving lane markings. Besides, one notices the distinct difference between the hypotheses, which only use the right or the left lane marking. For *LH, SLH*, and *TLH*, the most important features come from the corresponding left ego-lane markings and some of the consensus features. For *RH, SRH*, and *TRH*, only features concerning the right lane markings and features belonging to these hypotheses are ranked as important. The only exception can be found for *LH*.

Figure 18.14a–f shows that the features of the lane markings from the first camera and the second camera are sometimes mixed for the hypotheses *LH, RH, CH, SLH, SRH*, and *SCH*. The reason is because of the similar characteristics and the installation positions of both cameras. In contrast, only the lane markings received from the third camera and their belonging features are important for *TLH, TRH*, and *TCH* due to the different field of view.

Figure 18.15 shows that almost all features acquired from the leading vehicle are very relevant for *VH*. It is also interesting and correct that none of the marking information can be found here.

In summary, the main impact on the reliabilities of the hypotheses comes from the according detection source. Hence, a reliability estimator can be trained by using only the data of the corresponding detections. Furthermore, the observation that the first and second camera features are correlated indicates a strong redundancy between the cameras. For the evaluation of the classifiers, the neural networks from Sect. 18.4.5 are trained using the 25 highest ranked features.

18.6.3 Result of Reliability Estimation

To measure the classifier's performance, we use the $F_{0.8}$-Score which is defined as

$$F_\beta = \frac{(1+\beta^2)\cdot PR \cdot RC}{(\beta^2 \cdot PR) + RC} \tag{18.28}$$

where $PR = TP/(TP + FP)$ is the precision and $RC = TP/(TP + FN)$ is the recall. Moreover, TP denotes the number of true positive samples, FP the number of false positive samples and FN the number of false negative samples. The motivation of using a $F_{0.8}$-Score is that we want to increase the impact of the precision on the result and penalize false positives more than false negatives, since automated driving is a safety-critical application. The higher the $F_{0.8}$-Score, the better a classifier.

Figure 18.16 shows the classification results of ANN when predicting the reliability for the ten hypotheses. Since we perform a down-sampling on the evaluation data, there are the same numbers of reliable and unreliable samples. This is indicated by the *maximum availability*, i.e., the amount of positive samples over all samples, which is equal to 0.5 in most cases. Only for *FLH*, *FRH*, *SRH* and *SCH* from highway scenarios no down-sampling is needed, since all samples are positive. However, ANN estimates some hypotheses, such as *FRH*, *FCH*, *SLH* and *TCH*, to be reliable for all samples. This leads to a low $F_{0.8}$-Score of around 0.7.

For highway scenarios, the hypotheses *FLH*, *FRH*, *SRH*, and *SCH* have the best performance of around 100% (Fig. 18.16b). Following this, the performance for

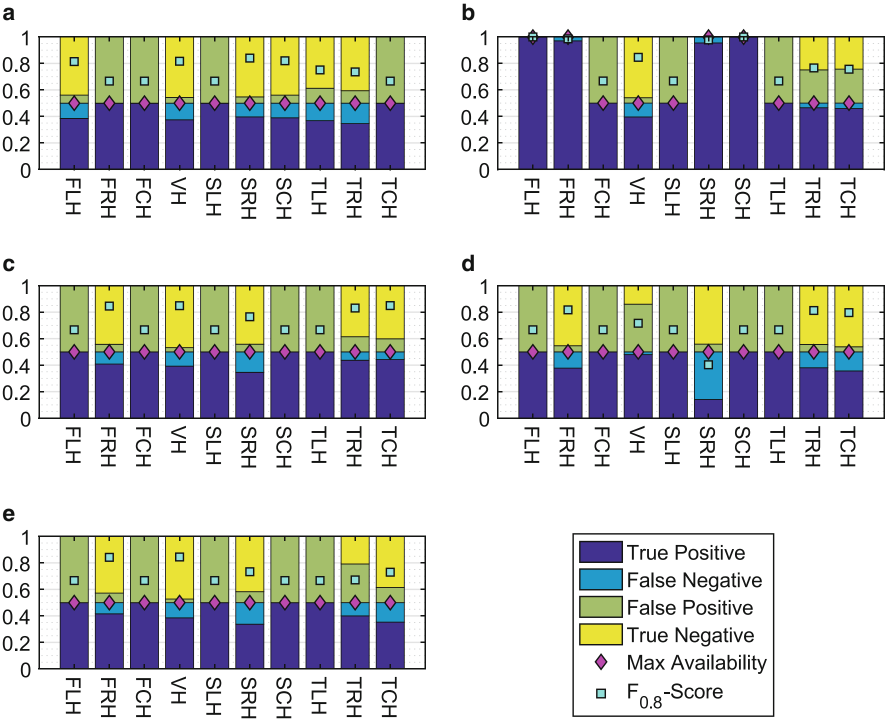


Fig. 18.16 Classification performance of ANN when predicting our ten hypotheses in different scenarios. (**a**) Overall. (**b**) Highway. (**c**) Rural. (**d**) Urban. (**e**) Connections

VH is about 80%. Moreover, ANN also performs well for *VH* in other scenarios. For rural scenarios in Fig. 18.16c, the classification performances for the ego-lane models based on the right lane markings are around 85%, which is better than the results of models based on the left lane markings. The center hypotheses *FCH* and *SCH*, which incorporate both lane markings, cannot improve their estimation result using the right markings and have the same bad classification performance as the left models *FLH* and *SLH*. Only *TCH* can make more use of both markings and is therefore ranked better than *FCH* and *SCH*. In urban scenarios, the performances of all classifiers decrease due to the variety of situations, where markings are sometimes not existing (Fig. 18.16d). In connection scenarios, all classifiers perform worse since the front-facing cameras cannot detect markings well here due to the narrow fields of view. It can be seen that *VH* has the highest performance in connection scenarios.

By using ANN as reliability estimators, the next section will evaluate the results of different fusion strategies from Sect. 18.4.6 and compare them with the direct ego-lane estimation approach from Sect. 18.5.

18.6.4 Result of Ego-Lane Estimation

The final estimation results are compared using two metrics: the commonly used lateral offset and the angle deviation from [28]. Both metrics are applied to the hypotheses at different run lengths to investigate the estimation quality both in close distance and in far distance to the vehicle.

As a general observation from Fig. 18.17, over 75% of the samples of each fusion strategy reach an angle difference of $\Delta\alpha < 2^\circ$ and a lateral offset of $\Delta d < 1\,\text{m}$. In the following, ANN denotes the fusion concept, where ANNs are used to directly estimate the parameters of the ego-lane. By comparing ANN with different fusion approaches, ANN turns out to perform well regarding short distances (Fig. 18.17a, b). However, both metrics agree that the error of ANN increases significantly as the distance grows. Thus, all fusion approaches outperform ANN after a run length of 28 m (Fig. 18.17g, h). The reasons for these results are the two design decisions when using ANNs for the direct ego-lane estimation process in Sect. 18.5. First, the usage of the polynomial for the ground truth acquisition induces an error, which is especially great in strong curves because the assumption of an angle below 15° does not hold. Second, the representation as a polynomial has the disadvantage of highly amplifying small mistakes in the estimation. For instance, if the ideal parameters are denoted by ϕ_0, C_0 and C_1 and the estimated parameters are denoted by $\tilde{\phi_0}$, $\tilde{C_0}$ and $\tilde{C_1}$, each estimated parameter can be written as

$$\tilde{\phi_0} = \phi_0 + \epsilon_{\phi_0} \tag{18.29}$$

$$\tilde{C_0} = C_0 + \epsilon_{C_0} \tag{18.30}$$

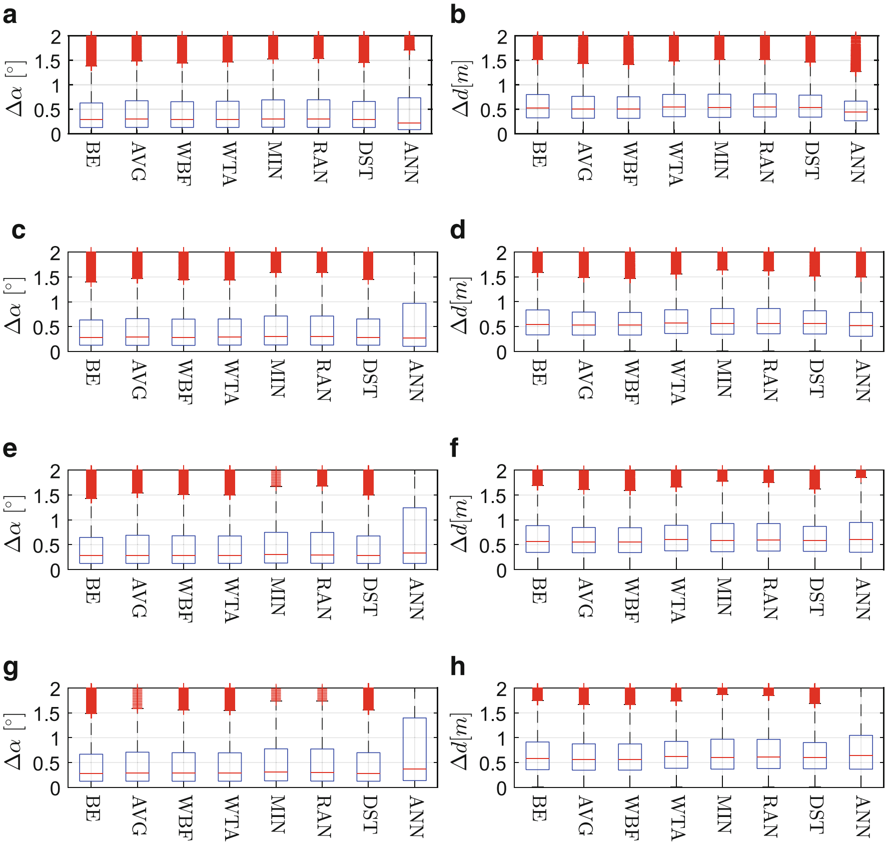


Fig. 18.17 Performance of different fusion strategies and ANN measured by the angle deviation $\varDelta\alpha$ and the lateral offset $\varDelta d$ to the ground truth at various distances. We excluded all samples with no reliable hypothesis. (**a**) Angle deviation at 16 m. (**b**) Lateral offset at 16 m. (**c**) Angle deviation at 22 m. (**d**) Lateral offset at 22 m. (**e**) Angle deviation at 28 m. (**f**) Lateral offset at 28 m. (**g**) Angle deviation at 31 m. (**h**) Angle deviation at 31 m

$$\tilde{C_1} = C_1 + \epsilon_{C_1} \tag{18.31}$$

where ϵ_p denotes the error in the estimation of parameter p. Next, the impact of the estimation error can be determined as the absolute error

$$e_{\text{abs}} = \left\| \phi_0 \cdot x + \frac{C_0}{2}x^2 + \frac{C_1}{6}x^3 - \left(\left(\phi_0 + \epsilon_{\phi_0}\right) \cdot x + \frac{C_0 + \epsilon_{C_0}}{2}x^2 + \frac{C_1 + \epsilon_{C_1}}{6}x^3 \right) \right\| \tag{18.32}$$

$$= \left\| \epsilon_{\phi_0} \cdot x + \frac{\epsilon_{C_0}}{2}x^2 + \frac{\epsilon_{C_1}}{6}x^3 \right\| \tag{18.33}$$

If the error at a distance of 31 m is considered and $\epsilon_{\phi_0} = 0$ and $\epsilon_{C_0} = 0$, the absolute error e_{abs} is $\left\| \left(4965 + \frac{1}{6}\right) \epsilon_{C_1} \right\|$. Hence, an error in C_1 greater than $0.00021 (\approx 0.012°)$ leads to a lateral offset of more than one meter. Analogously, the error in C_0 has a significant impact. For that reason, small errors in estimations can lead to poor performance of ANN with increasing run lengths.

To evaluate the final performance of the fusion strategies and ANN, we use the *availability (AV)*, which is given by the proportion of samples with a correct ego-lane estimation over all samples [23]. By that, a strategy is considered as available only when the following conditions are fulfilled. First, the strategy provides an estimate for the given sample. Secondly, the angle deviation $\Delta\alpha$ of the provided estimate must not exceed 2°.

For highways and rural roads, Fig. 18.18b, c show that the performances of all strategies are near 100% because of good road conditions in these scenarios. As expected, the performances of all strategies are lower in urban areas (Fig. 18.18d) due to the variety of situations. In on- and off-ramp scenarios, all strategies have their lowest availability (Fig. 18.18e). Compared to *BE* from [1], our fusion can enable an increase of up to 5 percentage points regarding the availability.

Furthermore, ANN has the lowest availability for all scenarios with the exception of connections. The overall low availability is expected looking at the performance regarding $\Delta\alpha$. Unfortunately, ANN performs even worse considering that AV is mostly smaller than MIN, which selects the hypothesis with the lowest reliability. In contrast, ANN achieves the best performance in connection scenarios. This is due to the weaker dependence on lane markings, which are hard to detect in curves. Hence, ANN can comprehend the lack of lane marking detection. Moreover, the results could be improved by using a different representation that suffers less from

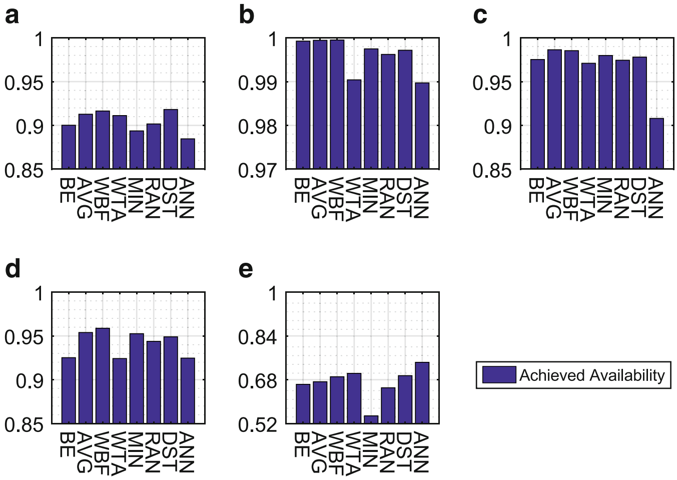


Fig. 18.18 Comparison of the achieved availability of different ego-lane estimation models in different scenarios. (**a**) Overall. (**b**) Highways. (**c**) Rural. (**d**) Urban. (**e**) Connection

errors in the prediction and the ground truth acquisition. As a consequence, the results could be used to improve the performance especially in curve scenarios.

18.7 Conclusion

In this work, we present two fusion concepts for ego-lane estimation by using multiple sensors and neural networks. The first approach estimates for each source a reliability value, which indicates whether the source is correct for the current situation or not. Based on the predicted reliabilities, the fusion will prefer reliable sources over unreliable sources, such as by giving greater weights to reliable hypotheses or by excluding unreliable sources from the fusion. Instead of explicitly estimating reliabilities, the second approach uses neural networks to directly estimate the ego-lane. Thereby, the reliabilities are internally learned and encoded as weights of the neurons. Compared to a standard road estimation approach from [1], our approach can increase the availability by up to 5 percentage points.

In future work, we want to improve both fusion concepts by changing the net structure and utilize different structures for different hypotheses and parameters respectively. Additionally, a further improvement of the feature selection needs to be done by comparing the performance of the same classifier using different features. The direct ego-lane estimation performs slightly worse than the results of other fusion strategies regarding the angle deviation and availability. However, the performance in connection scenarios is better than all other fusion approaches. For that reason, a possible use for ANNs would be to incorporate the estimation into the fusion framework and improve the performance in connection scenarios. When training ANNs, we found that the representation of the targets as an approximation of a clothoid is not appropriate due to the large amplification of errors in the estimation. Hence, a scalar field could be used instead, where the values above and below a threshold represent the lane. Furthermore, we plan to improve both neural network approaches by incorporating temporal information and using recurrent neural networks. This can lead to more sufficient estimations in all scenarios.

References

1. D. Töpfer, J. Spehr, J. Effertz, C. Stiller, Efficient scene understanding for intelligent vehicles using a part-based road representation, in *IEEE Conference on Intelligent Transportation Systems* (2013), pp. 65–70. https://doi.org/10.1109/ITSC.2013.6728212
2. T.T. Nguyen, J. Spehr, M. Uhlemann, S. Zug, R. Kruse, Learning of lane information reliability for intelligent vehicles, in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems* (2016), pp. 142–147. https://doi.org/10.1109/MFI.2016.7849480
3. T.T. Nguyen, J. Spehr, J. Xiong, M. Baum, S. Zug, R. Kruse, Online reliability assessment and reliability-aware fusion for ego-lane detection using influence diagram and Bayes filter, in *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems* (2017), pp. 7–14

4. T.T. Nguyen, J. Spehr, S. Zug, R. Kruse, Multi-source fusion for robust road detection using online estimated reliabilities. IEEE Trans. Indus. Inf. 1 (2018). https://doi.org/10.1109/TII.2018.2865582
5. C. Chen, A. Seff, A. Kornhauser, J. Xiao, DeepDriving: learning affordance for direct perception in autonomous driving, in *IEEE International Conference on Computer Vision* (2015), pp. 2722–2730
6. D.A. Pomerleau, Efficient training of artificial neural networks for autonomous navigation. Neural Comput. **3**(1), 88–97 (1991). https://doi.org/10.1162/neco.1991.3.1.88
7. M. Bojarski, D.D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L.D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, K. Zieba, End to end learning for self-driving cars (2016). CoRR abs/1604.07316
8. Z. Chen, X. Huang, End-to-end learning for lane keeping of self-driving cars, in *2017 IEEE Intelligent Vehicles Symposium (IV)* (2017), pp. 1856–1860. https://doi.org/10.1109/IVS.2017.7995975
9. F. Codevilla, M. Müller, A. Dosovitskiy, A. López, V. Koltun, End-to-end driving via conditional imitation learning (2017). CoRR abs/1710.02410
10. A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope. Int. J. Comput. Vis. **42**(3), 145–175 (2001). https://doi.org/10.1023/A:1011139631724
11. M. Al-Qizwini, I. Barjasteh, H. Al-Qassab, H. Radha, Deep learning algorithm for autonomous driving using googlenet, in *2017 IEEE Intelligent Vehicles Symposium (IV)* (2017), pp. 89–96. https://doi.org/10.1109/IVS.2017.7995703
12. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions (2014). ArXiv e-prints
13. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition (2014). CoRR abs/1409.1556
14. M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks (2013). CoRR abs/1311.2901
15. O. Hartmann, M. Gabb, R. Schweiger, K. Dietmayer, Towards autonomous self-assessment of digital maps, in *Proceedings of the IEEE Intelligent Vehicles Symposium* (2014), pp. 89–95. https://doi.org/10.1109/IVS.2014.6856564
16. G.L. Rogova, V. Nimier, Reliability in information fusion: literature survey, in *7th International Conference On Information Fusion* (2004), pp. 1158–1165
17. T. Brade, S. Zug, J. Kaiser, Validity-based failure algebra for distributed sensor systems, in *IEEE International Symposium on Reliable Distributed Systems* (2013), pp. 143–152. https://doi.org/10.1109/SRDS.2013.23
18. H. Frigui, L. Zhang, P. Gader, Context-dependent multi-sensor fusion for landmine detection, in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium* (2008), pp. II–371–II–374. https://doi.org/10.1109/IGARSS.2008.4779005
19. T.T. Nguyen, J. Spehr, J.-O. Perschewski, F. Engel, S. Zug, R. Kruse, Zuverlässigkeitsbasierte Fusion von Fahrstreifeninformationen für Fahrerassistenzfunktionen, in *Proceedings 27. Workshop Computational Intelligence*, ed. by F. Hoffmann, E. Hüllermeier, R. Mikut (KIT Scientific Publishing, Karlsruhe, 2017), pp. 33–49
20. T.T. Nguyen, J. Spehr, J. Sitzmann, M. Baum, S. Zug, R. Kruse: Improving ego-lane detection by incorporating source reliability, in *Multisensor Fusion and Integration in the Wake of Big Data, Deep Learning and Cyber Physical System*, ed. by S. Lee, H. Ko, S. Oh. Lecture Notes in Electrical Engineering, vol. 501 (Springer International Publishing, Cham, 2018)
21. M. Realpe, B.X. Vintimilla, L. Vlacic, A fault tolerant perception system for autonomous vehicles, in *Proceedings of the 35th Chinese Control Conference* (2016), pp. 6531–6536. https://doi.org/10.1109/ChiCC.2016.7554385
22. A. Rechy Romero, P.V. Koerich Borges, A. Elfes, A. Pfrunder, Environment-aware sensor fusion for obstacle detection, in *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems* (2016), pp. 114–121. https://doi.org/10.1109/MFI.2016.7849476

23. T.T. Nguyen, J. Spehr, D. Vock, M. Baum, S. Zug, R. Kruse, A general reliability-aware fusion concept using DST and supervised learning with its applications in multi-source road estimation, in *2018 IEEE Intelligent Vehicles Symposium (IV)* (2018), pp. 597–604
24. F.E. White, A model for data fusion, in *Proceedings of the First National Symposium on Sensor Fusion* (1988)
25. C. Gackstatter, P. Heinemann, S. Thomas, B. Rosenhahn, G. Klinker: Fusion of clothoid segments for a more accurate and updated prediction of the road geometry, in *13th International IEEE Intelligent Transportation Systems Conference* (2010), pp. 1691–1696. https://doi.org/10.1109/ITSC.2010.5625270
26. T.T. Nguyen, J. Spehr, H. Lin, D. Lipinski, Fused raised pavement marker detection using 2D-Lidar and mono camera, in *IEEE International Conference on Intelligent Transportation Systems* (2015), pp. 2346–2351
27. E.D. Dickmanns, B.D. Mysliwetz, Recursive 3-D road and relative ego-state recognition. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 199–213 (1992). https://doi.org/10.1109/34.121789
28. T.T. Nguyen, J. Spehr, J. Xiong, M. Baum, S. Zug, R. Kruse, A survey of performance measures to evaluate ego-lane estimation and a novel sensor-independent measure along with its applications, in *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems* (2017), pp. 239–246
29. A. Kraskov, H. Stögbauer, P. Grassberger, Estimating mutual information. Phys. Rev. E Stat. Nonlinear Soft Matt. Phys. **69**(6 Pt 2), 066138 (2004). https://doi.org/10.1103/PhysRevE.69.066138
30. J. Pradeep, E. Srinivasan, S. Himavathi, Diagonal based feature extraction for handwritten character recognition system using neural network, in *2011 3rd International Conference on Electronics Computer Technology (ICECT)* (2011), pp. 364–368. https://doi.org/10.1109/ICECTECH.2011.5941921
31. A. Graves, A.R. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2013), pp. 6645–6649. https://doi.org/10.1109/ICASSP.2013.6638947
32. R. Kruse, C. Borgelt, C. Braune, S. Mostaghim, M. Steinbrecher, *Computational Intelligence: A Methodological Introduction*. Texts in Computer Science, 2nd edn./2016 edn. (Springer, London, 2016)
33. C.M. Bishop, *Pattern Recognition and Machine Learning*. Information Science and Statistics (Springer, New York, 2006)
34. L. Bottou, Stochastic gradient descent tricks, in *Neural Networks: Tricks of the Trade*, ed. by G. Montavon, G.B. Orr, K.R. Müller, 2nd edn. (Springer, Berlin/Heidelberg, 2012), pp. 421–436
35. L. Bottou, Stochastic gradient learning in neural networks. Proc. Neuro-Nımes **91**(8), 687–696 (1991)
36. N. Qian, On the momentum term in gradient descent learning algorithms. Neural Netw. **12**(1), 145–151 (1999). https://doi.org/10.1016/S0893-6080(98)00116-6, http://www.sciencedirect.com/science/article/pii/S0893608098001166
37. I. Sutskever, J. Martens, G. Dahl, G. Hinton, On the importance of initialization and momentum in deep learning, in *Proceedings of the 30th International Conference on Machine Learning – Volume 28*, ICML'13 (2013), pp. III–1139–III–1147. http://dl.acm.org/citation.cfm?id=3042817.3043064
38. G. Shafer, *A Mathematical Theory of Evidence* (Princeton University Press, Princeton, 1976)
39. M. Aeberhard, S. Paul, N. Kaempchen, T. Bertram, Object existence probability fusion using dempster-shafer theory in a high-level sensor data fusion architecture, in *Proceedings of IEEE Intelligent Vehicles Symposium* (2011), pp. 770–775. https://doi.org/10.1109/IVS.2011.5940430
40. A.E. Albert, A.L. Albert, *Regression and the Moore-Penrose Pseudoinverse*. Mathematics in Science and Engineering: A Series of Monographs and Textbooks (Academic, New York, 1972)