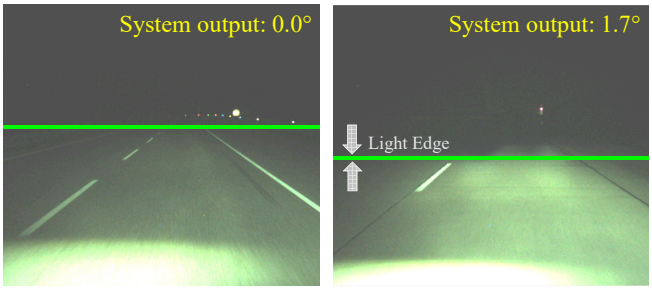
# Headlight Range Estimation for Autonomous Driving using Deep Neural Networks

Jakob Mayr[*1,2], Can Giracoglu[*2], Christian Unger[1], Federico Tombari[2]

***Abstract*— When driving at night, a good illumination of the road ahead is crucial. With autonomous driving at close temporal proximity, this not only concerns human drivers but also autonomous systems capable of controlling the car. To achieve fully autonomous driving, a variety of sensors are integrated into the vehicles. Cameras act as one of the major sensors. However, due to their passivity, cameras cannot see well in the dark. To mitigate this shortcoming, modern cars are equipped with powerful headlights that provide proper illumination of the road ahead while avoiding the dazzling of other traffic participants. To use the headlights' full potential and to also provide advanced light functionality like glare-free high beam, they need to be properly adjusted. After the initial calibration during production, this setting is prone to undesirable degradation, primarily due to mechanical reasons. We present a completely new application of computer vision and machine learning to automatically detect wrongly adjusted headlights by estimating their pitch angle from the images of a vehicle-attached camera for advanced driving assistance systems (ADAS). We show that we can achieve high performance in terms of accuracy and robustness by training a deep neural network in an end-to-end fashion. To demonstrate the benefits of our proposed approach, an additional handcrafted baseline method is implemented.**

## I. INTRODUCTION

Modern vehicle headlights provide optimal illumination of the road ahead in the dark to facilitate the recognition of the environment, e.g. road, obstacles or traffic signs. However, headlight design has to compromise between the maximum headlight range and requirements that no traffic participants are dazzled. Each headlight is adjustable and has two degrees-of-freedom (DoF), yaw and pitch (see Fig. 2) influencing the lateral orientation and longitudinal range of the light cone, respectively. During assembly of the vehicle or in customer service, both DoF are checked and set correctly. These settings provide optimal illumination for the driver as well as for ADAS cameras installed to the vehicle and are also a prerequesite for modern light functions like glare-free high beam. Headlight orientation degrades over time because of time-related mechanical issues, though. Therefore, it has to be manually checked and re-adjusted after certain time intervals. We propose an entirely new application of computer vision to avoid any manual adjustments by processing monocular ADAS camera images with a deep neural network (DNN) that estimates the pitch angle of the headlight. An exemplary result of our promising approach can be seen in Fig. 1, where we can see an image of optimal headlight setting (a) and of a deteriorated setting (b). Our system prototype in both cases correctly estimates the pitch angle, thus providing a possibility to automatically readjust the headlight.




(a) Optimal setting (0°) (b) Deteriorated setting (1.7 °)

Fig. 1: Two exemplary headlight pitch settings and the corresponding estimations of our approach for (a) a correctly adjusted and (b) misaligned headlight. The proposed system correctly and robustly detects the misalignment

Our attention will be on pitch angle changes since this angle is more prone to degradation. Additionally, the pitch angle has more impact on the illumination of the road as well as on possible dazzling of other traffic participants. To estimate the pitch angle from the images, end-to-end deep neural networks are trained to model the complex physical relationships between headlight settings and road characteristics. We compare the effectiveness of a handcrafted baseline method incorporating binocular vision depth information and diverse deep learning architectures. In a first development step, the complexity of headlight range estimation is slightly reduced by ensuring defined environment and vehicle conditions in the dataset:

1) No other light sources in the camera image (headlights of other cars / street lights).
2) Homogeneous road surface.
3) Good weather conditions and dark night.
4) Low vehicle dynamics (lateral/longitudinal vehicle acceleration reduced to a minimum).

This enables a quick proof-of-concept and also facilitates the analyses of impact factors that deteriorate the estimation results. Additionally, all named factors are easily detectable by current series-production ADAS systems, and the degradation of the headlight is no dynamic process but slowly occurs over time. Hence, during normal driving, it is possible

[1]BMW Group, 80788 Munich, Germany, jakob.mayr@bmw.de, christian.unger@bmw.de
[2]Chair for Computer Aided Medical Procedures, Technical University Munich, Boltzmannstr. 3, 85748 Garching, Germany, tombari@in.tum.de, can.giracoglu@tum.de
[*]Equal Contribution

to wait until situations occur that satisfy above conditions and estimate only then to get a reliable estimation. However, as our evaluation shows, even when training on scenes satisfying above conditions, estimations during normal public road driving yield promising results especially when using temporal-filtering techniques.


Fig. 2: Visualization of Yaw and Pitch rotation DoFs of the headlight

The contributions of this paper can be summarized as follows:

1) Implementation of a traditional non-learning computer vision approach to automatically detect headlight pitch angles from ADAS camera images.
2) Implementation and training of a deep learning approach to learn to estimate the headlight setting in an end-to-end fashion.
3) An evaluation and comparison of the proposed methods as well as different deep neural network architectures (DNN).
4) An investigation focusing on the limitations of the current prototype and the most impacting factors by recording an additional public road dataset.

## II. RELATED WORK

There has not been any work published on the problem of estimating the headlight pitch angle or range from ADAS camera images (or any other sensor) so far. In that sense, we are working on an entirely new application. Nevertheless, we are using tools that are well known and summarized in the following. It should be noted, that although the presented work focuses on the task of range or depth estimation it also applies to pitch angle estimation since it is only another representation of the same circumstance.

### *A. 1-Dimensional Range Estimation using Machine Learning*

1-Dimensional range estimation is not a heavily investigated topic in the field of machine and deep learning in contrast to complete depth map estimation [1] which is an intense research interest and can be seen as an alternative to our 1-D approach. However [2] follow a similar approach to solve the problem of visibility range estimation under foggy weather conditions by trying to develop an intelligent fog detection systems using neural networks. They extract global descriptors of images using Fourier transform and Shannon entropy in a separate pre-processing step. Then these features vectors are fed into a shallow 3-layered neural network where the output layer is a vector having the size of the number of classes. In other words, the problem is considered as a classification problem, and each class is considered as certain range intervals (e.g. 60-100m is class 2). Apart from the different use-case, our approach differs in the sense that we train a deep neural network in an end-to-end fashion solely using the images as input and having the headlight pitch as output. Although our labels are discrete angle steps, we at last formulate the problem not as a classification but as a regression task. This fits our problem much better because a regression loss also considers how far from the ground truth an estimation is instead of just penalizing that the result is wrong, e.g. if the pitch angle is actually $0.8°$ an estimation of $1.5°$ is worse than an estimation of $1.0°$.

### *B. 1D Range Estimation using Traditional Computer Vision*

*1) Depth Estimation Using Binocular Vision:* Depth information from stereo vision can be extracted by matching each pixel in the left image with its right image correspondence which results in the disparity (i.e. inverse depth) map [3]. There is a relationship between the real-world depth $Z$ and image disparity $d$ where the baseline $b$ is the distance between the two stereo camera lenses (in meters) and $f$ is the focal length (in pixels) [4]:

$$Z = \frac{b * f}{d} \tag{1}$$

In our case, stereo matching is a challenging task due to two reasons: First, the environment is very dark due to night time driving making it very hard to match pixels from dark areas not illuminated by the headlight. And secondly, we primarily face a uniformly textured road surface where easily matchable features are missing for the most part.

*2) Simplified Monocular Depth Estimation Utilizing a-priori Camera Information:* When working with monocular vision, depth information is in general only recoverable up-to-scale. To get the absolute depth a-priori information needs to be fused in. One method to recover depth involves the knowledge of camera height, pitch angle, the field of view, focal length and only works under the assumption of a planar road surface. In [5], the authors address the problem of inter-vehicle distance estimation. To do so, a birds-eye-view representation of the road is constructed. It enables them to estimate the real-world distance between two objects after finding pixel distances between objects and comparing them with a ground truth metric for the same camera setup and parameters. However, they show that the interpolation errors exponentially increase for distances larger than 30 meters in the ”bird’s eye view” approach. Another method for distance estimation utilizes perspective camera modeling as in [6]. This model uses relationships between triangles in the real world and image space to relate distances between an object on the image (in pixels) to the real world (in meters). The weak side of this algorithm is that it is sensitive to bumps or any vertical movement and can hence not be considered a robust solution. Due to the

higher accuracy and robustness, we solely focus on depth information extraction from stereo vision, for the baseline method while our DNN approach solely uses monocular images. It should be noted that additionally to estimating the depth, first the development of a light edge detection is required in the baseline approach as discussed in section III-B.

## III. METHOD

For the implementation of a handcrafted baseline method, we utilize the stereo depth map. However, our proposed method for estimating the headlights' pitch orientation is a monocular end-to-end deep learning approach.

### A. Converting depth values to pitch angles

Our training images are labeled with the corresponding pitch angle of the headlights. For the baseline method, we estimate the headlight range in meters and then convert them to degrees to allow for a consistent evaluation. We use the known relative geometry of the camera and headlights to establish a mapping function that associates the range space in meters with the pitch angle space in degrees.

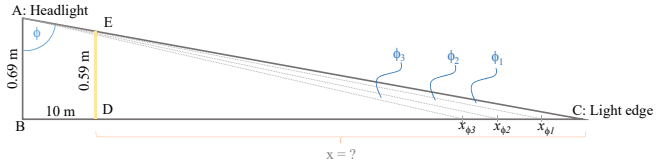



Fig. 3: Relationship between pitch angle and headlight range information

As visualized in Fig. 3, we measure three distances to enable the conversion of angle and range labels:
$\overline{AB}$ is the height of headlight from the ground.
$\overline{BD}$ is the distance to a wall.
$\overline{DE}$ is the height of the light edge on the wall from the ground.
We exploit the distance ratios between triangles $\overset{\triangle}{ABC}$ and $\overset{\triangle}{CDE}$ to determine the headlight range at the initial pitch angle setting (i.e. relative angle of $0°$) to be $\overline{BC} = 69m$. We can also compute the absolute angle of the headlight $\phi$:

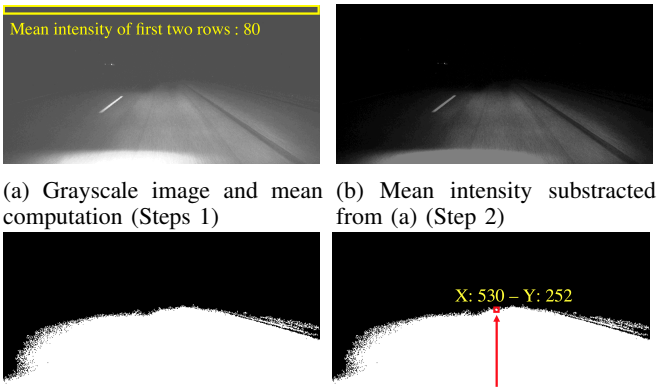
$$\phi = arctan(\frac{0.69}{69}) = 0.57° \tag{2}$$

Hence, for the relative angle $0°$ the absolute angle of the headlights is $0.57°$.

### B. Baseline Light Edge Detection Algorithm

In order to estimate the headlight range and consequently pitch angle in the non-learning baseline approach, we first need to extract the light's edge on the road (see Fig. 1) Please see Fig. 4 for some visual insight on the different steps that are described here:

1) Get the mean pixel intensity of the top two rows of the grayscale image.
2) Subtract this mean intensity value from all pixels in the image.
3) All pixels with a remaining intensity exceeding 0 are labeled with a 1 and the others are set to 0 (resulting in a mask).
4) Apply median filter to remove salt and pepper noise.
5) Select the largest connected component in the mask.
6) Find the centroid of this area and get its X coordinate.
7) Obtain the area's topmost Y coordinate on the X column.




(a) Grayscale image and mean computation (Steps 1)

(b) Mean intensity substracted from (a) (Step 2)

(c) Denoised illumination mask (Steps 3-5)

(d) Identified light edge reference point (Steps 6&7)

Fig. 4: Sample images visualizing various steps of the baseline light edge detector

### C. Baseline Depth estimation using Binocular Vision

We recorded our dataset at night resulting in dark images. Consequently, the quality of the pixel matching process is low and the disparity map is quite noisy especially in the darker regions. Therefore, the masking algorithm explained in the previous section is used to focus on bright regions instead (we also try to detect lane markings as they are bright and result in reliable disparity values, however solid markings are not always available). After detecting the light edge with the described algorithm, the corresponding depth values are extracted and converted into angle values to allow evaluation with the ground truth. The algorithm can be summarized as follows:

1) Apply stereo matching on grayscale image pair.
2) Apply illumination mask to the disparity map.
3) Compute depth from the disparity map.
4) Apply a simple gradient-based method to detect lane markings.
5) If lane markings are detected:
   a) Filter the depth map with the lane marking mask.
   b) Determine the depth of the lane marking on the light edge's Y-position.
6) If no lane markings are present or detected, determine depth of the light edge reference point (see 4 (d)).
7) Convert the headlights' range into the corresponding pitch angle.

There are several error sources in this simple baseline method. Firstly, in our implementation, the range is calculated based on a single disparity. This leads to noisy results especially since it is in many cases challenging to detect the light edge. Secondly, assuming a good light edge detection, the stereo depth estimation approach is vulnerable for wrong estimations in dark areas. For the baseline we therefore use the median of the last ten estimates to prevent unstable results.

### *D. Investigating Deep Learning-Based Methods*

We carry out an evaluation of different DNN architectures including an ablation study in order to find the best performing network for our application and also to understand the impact of different environmental properties, e.g. the effects of a curved road or rainy weather, on the performance. The model is tested on two datasets: 1) dataset with similar environmental conditions as the training data but recorded on another day and 2) a more challenging dataset recorded in a completely different environment on public roads. Each training is run for ten epochs with an initial learning rate of $0.0003$ (dropped by a factor of ten after seven epochs). Furthermore, we use a minibatch size of 16 and L2-Regularization of $1e-08$ using the Adam optimizer. To diversify the training data we decided to incorporate data augmentation techniques. We use random rotations ($\pm 10°$) and random X and Y translations within a pixel range $\pm 4px$. When formulated as a classification task we use "Softmax" and for regression we choose the "Mean Squared Error" as loss functions. Please see Table II for an overview of all tested architectures.

### *E. Dataset for headlight range estimation*

In total, we recorded six data subsets on different days and acquired 23 hours of data. The first five subsets are gathered during the night on a closed test area where there is neither traffic nor other light sources. The recording of data on a non-public road allowed us to freely adapt the headlight pitch in a large range including severe misalignment that would not have been possible on public roads. The headlights' pitch angles are changed relative to the initial angle setting in a range between $0°$ and $1.7°$ with a resolution of $0.1°$ resulting in 18 different settings. This data can be regarded as ideal and allows for a first proof-of-concept. If the prototype is not capable of delivering high-performance predictions on this data further investigations on real driving scenarios do not make any sense. Involuntary, this data already contains significantly changed environment characteristics due to rain. This especially changed the reflectivity of the road surface which in turn shifted the perceived location of the light edge. The data also contains minor variance from non-homogeneity of the road surface texture, curves and slight pitch movements of the car. Parameters such as yaw rate or velocity are used to filter the dataset. For instance, straight and curved road sections were separated using the yaw rate of the car. After all, we created a dataset that almost contains one million valid images. Additionally, we recorded another verification dataset on public roads featuring more realistic scenarios. This data contains driving in rush-hour urban traffic, driving on a highway with other cars present and also rural road sections. The primary aim of this dataset is to analyze limitations of the proposed system and their different impact factors (e.g. oncoming vehicles, curves, non-vehicular lights, sloped road).

## IV. DNN EXPERIMENTS

### *A. Selection of a suitable DNN architecture and training procedure*

Conditions: All the following experiments are carried out on straight road sections at dry weather conditions. Each experiment in this section is repeated 20 times.

*1) Training from-scratch vs. transfer learning:* Ideally, the network should be trained with our own data only. However, when training the networks from scratch severe overfitting is an issue supposedly due to a lack of variance in the data. We hence use pre-trained networks that have been trained on the ImageNet [7] dataset. The promising results for the pre-trained networks, see Table I, prove that this is highly beneficial for our task with an accuracy improved by a factor $> 3$. The reason is that they are trained over a wide variety of several million images, which makes them capable of extracting more general features from the images. Hence, for the remaining experiments, pre-trained networks are used.

TABLE I: Mean accuracy and standard deviation for from-scratch training and pre-trained networks with ResNet101 [8] (Regression)

| ResNet101 [8] , Regression<br>Data: Dry, Straight | Mean Accuracy | Std. |
|---|---|---|
| From-Scratch | 0.1281 | 0.0437 |
| Transfer Learning | **0.4431** | 0.0390 |

*2) Comparison of different pre-trained DNN models:* Another major impact factor on the performance of a DNN is the choice of a suitable architecture. We therefore compare the performance of several DNNs. In our experiments the ResNet [8] architectures outperform other state-of-the-art model. The detailed results are listed in Table II. ResNet101 almost triples the accuracy of the next best network. It should be already noted here that although we use the accuracy as a single valued measure to determine a models performance, this does not reflect its full capabilities. As the confusion matrix shows (Fig. 7), wrong estimations usually have an error in a range of $\pm 0.1°$, which is a quite accurate result even though the accuracy is "only" $44.31\%$.

Looking at Table II we can also see that all models train relatively stable with a standard deviation of multiple trainings of one to five percent.

*3) Classification vs. Regression loss function:* Due to the step function used to change the headlight pitch angle during recording of the dataset we have a set of 18 discrete labels. It is therefore possible to formulate our problem either as a classification or a regression task. The natural choice is the

TABLE II: Comparison of different network architectures (utilizing pre-training on ImageNet [7])

| Pre-Trained, Regression<br>Data: Dry, Straight | Mean Accuracy | Std. |
|---|---|---|
| SqueezeNet [9] | 0.0379 | 0.0121 |
| AlexNet [10] | 0.1500 | 0.0357 |
| GoogleNet [11] | 0.1676 | 0.0303 |
| InceptionV3 [12] | 0.1554 | 0.0528 |
| ResNet50 [8] | 0.3919 | 0.0532 |
| ResNet101 [8] | **0.4431** | 0.0390 |

formulation as a regression task because the different angles are not independent classes but related. Regression not solely relies on a binary correct/wrong decision but also considers the distance between prediction and ground truth. To confirm our assumptions, we execute an experiment . Table III shows the result for this comparison. As expected the regression loss outperforms the classification loss for our dataset with the accurac almost doubling in value. Hence, the remaining experiments incorporate regression loss.

TABLE III: Results for classification loss compared to regression loss

| Pre-Trained ResNet101<br>Data: Dry, Straight | Mean Accuracy | Std. |
|---|---|---|
| Classification loss | 0.2207 | 0.0528 |
| Regression loss | **0.4431** | 0.0390 |

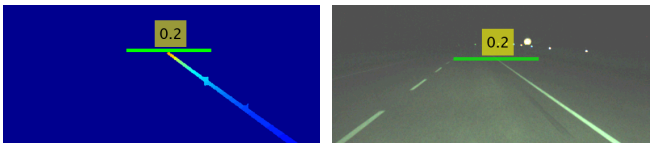
### B. Selection of training input data

For the following experiments ImageNet pre-trained ResNet101 is used, and the task is formulated as regression.

*1) Different color spaces and additional input information:* This experiment is carried out to see whether different color spaces or any additional information extending the 3-channel RGB color input is beneficial. We test RGB against CieLab and HSV color space. The intuition is, that the latter colorspaces with separate illumination channel ease the identification of bright and dark image regions for the network. Additionally we enrich the input of the network by feeding (additionally to RGB data) the baseline's illumination mask (as explained in Section III-B) and the stereo depth map to the network. The illumination mask might give an additional hint about the location of the light edge. The depth map on the other hand is fed as a helping indicator for real-world depth (and directly connected to the pitch angle). Unfortunately, neither of these ideas proved to be beneficial and even deteriorate the result as can be seen in Table IV. The best performance is still observed with RGB and RGB+Mask input. The main reason for lower accuracies of other color spaces is probably that the pre-training of the networks on ImageNet are executed on RGB images. Therefore, they cannot entirely benefit from the learned features when the color space is changed. In the case of RGB+Depth the poor quality of the created depth maps (especially in dark image regions) seems to not add useful information.

TABLE IV: Different input modalities and their influence on the performance

| Data: Dry, Straight | Mean Accuracy | Std. |
|---|---|---|
| RGB (3Ch) | **0.4431** | 0.0390 |
| CieLab (3Ch) | 0.2989 | 0.0435 |
| HSV (3Ch) | 0.2937 | 0.0816 |
| RGB-Mask (4Ch) | 0.4283 | 0.0526 |
| RGB-Depth Map (4Ch) | 0.1870 | 0.0274 |




(a) Baseline estimation (b) DNN estimation

Fig. 5: Sample estimation outputs of a) the baseline method and b) the DNN approach. In this case, both correctly predict a pitch angle of $0.2°$

*2) Weather conditions: Dry vs. Rain:* Although our primary dataset is recorded in a defined environment it contains data recorded during dry and during rainy weather. To show the severe appearance change between these conditions and their influence on the performance we train two model instances separately on dry and rainy data and then test each on both conditions. We observe a significant performance drop when a model trained on dry data is tested on rainy conditions and vice versa. Please see Table V for detailed results. The obtained results are close to random. This gives an impression of the strong impact of weather conditions. We also notice that training and testing on rainy data influences the performance negatively with the accuracy dropping by more than $60\%$.

TABLE V: Cross validation of training on dry/rainy data and testing on dry and rainy data

| Data: RGB, Straight | Mean Acc. | Std. |
|---|---|---|
| Train on Rainy / Test on Dry | 0.0862 | 0.0231 |
| Train on Rainy / Test on Rainy | **0.1777** | 0.0186 |
| Train on Dry / Test on Dry | **0.4431** | 0.0390 |
| Train on Dry / Test on Rainy | 0.0548 | 0.0620 |

*3) Curved Road vs. Straight Road vs. All Road:* As explained we exclude all images recorded in curves to make the conditions as uniform as possible. Since the approachs general feasibility is shown in above experiments we explore the influence of curves. Again, we notice an expectable drop in performance when cross-validating networks trained on only straight road images and testing on curvy road segments (see Table VI. However, when trained on a heterogeneous dataset containing both road types the network slightly improves compared to the best straight-only model. The reason for that might be a lower risk of overfitting because the training data is diversified.

In conclusion we observe that from the many impact factors, weather conditions have the most substantial influ-

TABLE VI: Influence of training on datasets containing straight-only, curve-only and mixed road sections

| Data: RGB, Dry | Mean Acc. | Std. |
|---|---|---|
| Train on Curved / Test on Straight | 0.3583 | 0.0710 |
| Train on Straight / Test on Curved | 0.3760 | 0.0537 |
| Train on All / Test on Curved | **0.4973** | 0.0613 |
| Train on All / Test on Straight | 0.4643 | 0.0517 |
| Train on All / Test on All | 0.4661 | 0.0464 |

ence on the results. We therefore train three separate model instances: *rainy model*, *dry model* and *mixed model* (trained on rainy and dry data). All data is selected from straight and curvy road sections. Based on the extensive ablation study we are able to identify ResNet101 [8] as the most suitable architecture for our application. We incorporate pre-training on ImageNet [7] and formulate our problem as a regression problem. A hyper-parameter optimization is conducted to select the best collection of parameters. Until now, we stated mean accuracies from multiple trainings per model to also show how stable and reproducible the training is. However, when it comes to the actual application of a network we obviously choose the single best performing model and perform the final evaluation on that model only.

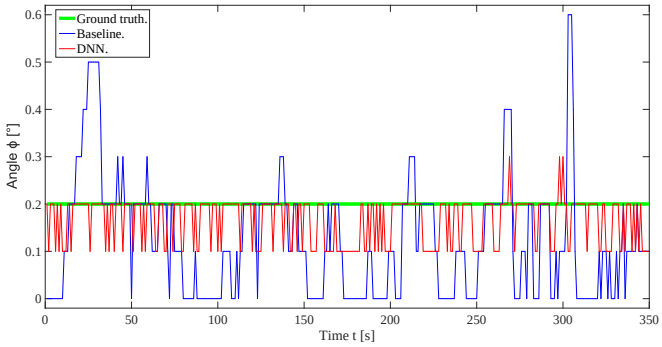



Fig. 6: Prediction over time of the baseline and the DNN approach

## V. RESULTS

### *A. Comparison of the hand-crafted baseline and the deep learning approach*

With the impressive results that deep learning methods achieve in various challenging problems there is a tendency to solve every task with the help of giant datasets in combination with state of the art networks. In some cases it is possible to solve the problem solely with traditional methods, though. This is one of the reasons why we also implement a non-learning baseline method. It enables us to evaluate if collecting annotated data and training of a DNN is reasonable and gives better results. Another reason is, that the presented application is new and there has not been any publication on this topic we can compare to. To rank the performance of the proposed learning approach it is very useful to have this baseline implementation. It is noteworthy, that we did not focus to solve the problem without deep learning, though and hence invested limited time on the baseline solution. When comparing the two approaches we first carefully select a sample path in order to satisfy conditions that are optimal for the baseline solution, i.e. where the light edge is always detected correctly and lane markings are present and also detected by the algorithm. The lane marking prerequisite is introduced to help the baseline to utilize the relatively reliable depth values deriving from lane markings in contrast to the noisy values in dark image areas. Additionally, the reason why several conditions are set in favor of the baseline approach is that when applying such an algorithm in the vehicle, it is possible to circumvent bad conditions by incorporating the output of numerous other vehicle signals, e.g. the output of lane detection or oncoming vehicle detection. Images are selected from dry data only, and hence the dry model is used for prediction. Fig. 5 shows two sample estimations of the baseline (a) and of the DNN method (b). On this optimal path, the baseline approach achieves an accuracy of $29\%$ and a mean absolute error (MAE) of $0.12°$. As we can observe from Fig. 6, despite the careful scene selection in favor of the baseline, the baseline implementation (in blue) does not yield a robust estimation result with many severe outliers. In contrast our deep learning approach performs well. The accuracy increases to $63\%$ and the output (in red) is quite stable, only slightly oscillating around the ground truth resulting in a MAE of $0.04°$. Of course, the baseline method's building blocks can be further improved but in the end the variety in the input data that we are facing will be always problematic. In our opinion, it is not reasonable to try to cover these countless scene properties with a hand-crafted method. In this task the deep learning method exploiting a vast amount of training data clearly is the more promising approach. To further investigate the robustness and to show that we do not overfit on the training and testing data only recorded at defined test track conditions we also evaluate the DNN approach on a public road dataset (please refer to Section V-C). Since the baseline method already shows very noisy results on the cherry-picked data, we do not further evaluate this approach.

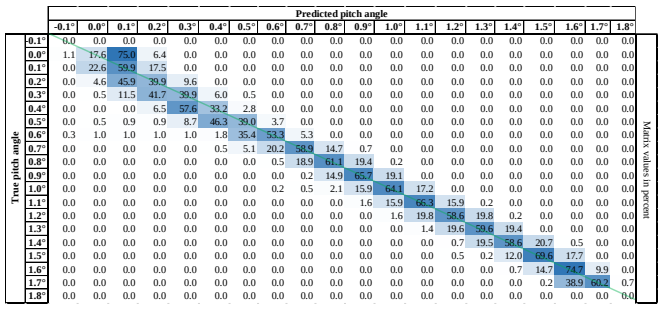



Fig. 7: Confusion matrix of the best performing model (mixed model, tested on dry straight&curved) (Best viewed as PDF)

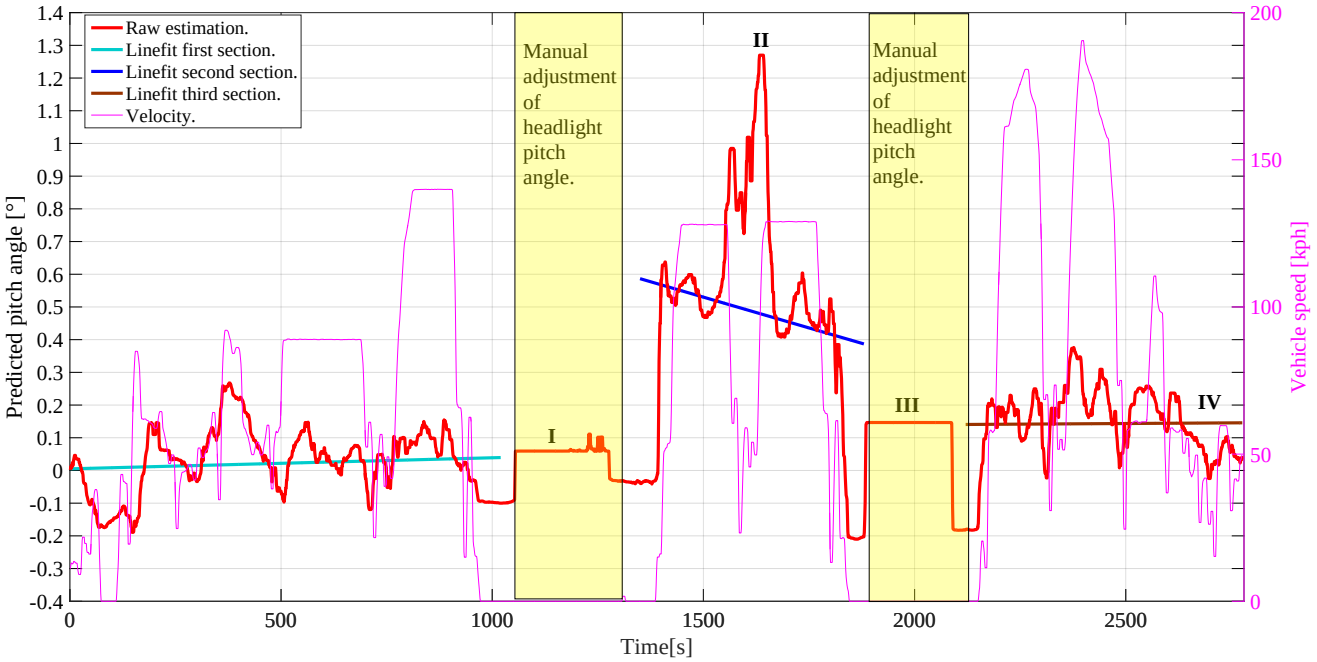



Fig. 8: Estimation result on validation dataset including vehicle velocity and linear model fit recorded on public roads. Roman numbers on the graph indicate special occasions shown in Fig. 9

### B. Final Evaluation of the DNN Approach

When evaluating the three model instances *dry, rainy, mixed*, the mixed model trained on rainy and dry data achieves the best performance on both, the dry and the rainy&dry testsets with accuracies of $57.90\%$ and $39.98\%$, respectively (see Table VII). We also observe that rainy environment conditions are not optimal with an accuracy of just $29.57\%$ when predicting with the rain-conditioned rainy model and slightly worse $24.21\%$ with the universal mixed model. It seems to be advisable to either collect more training data at rain or to not use rainy drives for an automatic estimation at all. This would not present a major problem since rain can be easily detected by modern cars already.

TABLE VII: Comparison of the best model accuracies (after hyper-parameter optimization) on different testsets

| Tested Data Features | Dry Model | Rainy Model | Mixed Model |
|---|---|---|---|
| Rainy | 0.0799 | **0.2957** | 0.2421 |
| Dry | 0.5676 | 0.1455 | **0.5790** |
| Mixed | 0.3124 | 0.2210 | **0.3998** |

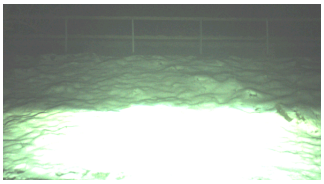
When only assessing the accuracies one might not be fully convinced about the usefulness of the presented approach since an accuracy of roughly $58\%$ also means that the model is wrong in $42\%$. However, the accuracy does not reflect the full performance of the estimation due to the inherent binary right/wrong evaluation. The confusion matrix is better suited for an in-depth assessment of the performance. With the y-axis being the true angles and the x-axis being the predicted angles of a test set, a perfect model would have only zeros except for the diagonal (i.e. every sample correct). Fig. 7 shows the confusion matrix for the mixed model tested on dry data. The color-coding immediately reveals that the cells on and around the diagonal are of high value. Despite an accuracy of only $57.90\%$, we observe that predictions more than $\pm 0.1°$ off are seldom. Predictions get more scattered as the angle gets closer to $0°$, though. The reason for that lies in the distinguishability of the light edge on the road. The closer it gets (i.e. the higher the misalignment of the headlights), the more obvious it is visible on the road resulting in more reliable network outputs. But even in the lower angle ranges the network is only off by $-0.2°$ in the worst case samples. Hence, we achieve a low overall MAE of only $0.05°$.

### C. Real Validation Scenario

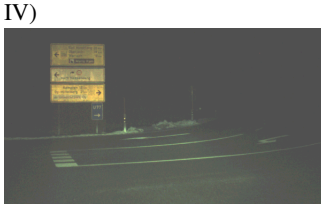
The robustness of a DNN is always an important aspect that needs to be assessed. We train our network on data that has been recorded at defined conditions only possible by recording on a closed test track. The testsets are also from the same roads (recorded on other days though) and hence we have a potential risk of overfiting on this data. We therefore record a completely independent dataset on public roads, where there are more adversarial and unfamiliar conditions to validate the model's robustness and give an insight on its limitations. We feed the images to the network without any pre-processing. However, that data severely differs from the training data and we introduce a simple temporal filtering by fitting a linear model to get an overall trend. The dataset is primarily recorded with a pitch angle of $0.0°$. We change the angle for two short driving periods to check if the network is capable of detecting such changes. The first change is significantly different from $0.0°$ wheras the second change is close to the initial setting. As we can see in Fig. 8, the inital angle as well as both changes are properly detected

by the DNN despite the new environment on a public road. Especially the second change to an angle close to the initial angle setting shows the usefulness of our approach as minor degradation can already be detected. Interestingly, the vehicle velocity does not have the expected strong impact as we can also determine from Fig. 8. The vehicle speed reaches up to $180km/h$ but the estimation result keeps stable. We also see that the model fitting works better when it processes more single estimations.

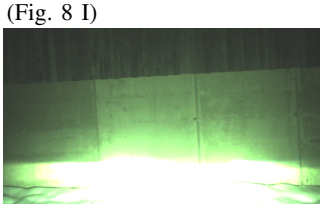
(a) External light sources (Fig. 8 IV)



(b) Failure mode: Strong slope (Fig. 8 I)



(c) Failure mode: Cluttered scene (Fig. 8 II)



(d) Failure mode: Large obstacle. (Fig. 8 III)

Fig. 9: Challenging conditions in the validation dataset recorded on public roads

We also evaluate our model on a 90 minute drive with correct $0.0°$ setting including urban and highway rushhour driving which confirms the performance seen in Fig. 8 with a MAE of $0.07°$ on the single outputs and $0.05°$ with temporal filtering. Again, this result is especially promising since estimations are more difficult around the $0.0°$ setting as already discussed in Section V-B.

Of course, there are also some problematic situations causing outliers that we will address in the following.

*1) Limitations of our current DNN prototype:* Our model proves to be very useful when testing on similar conditions and even shows generalization capabilities on harder public road test sets. However, especially due to the flawless training data we are able to identify some typical situations (apart from the already elaborated problems of weather and straightness of the road) where our current prototype model is challenged by:

- Obstacles: We train the DNNs on open view conditions without traffic, so any obstacle in front of the car is a potential cause of degradation. E.g. traffic signs located in front of the car (see Fig. 9 (c)) result in an observable estimation change. Furthermore, by looking at Fig. 9 (d), it can be stated that the size of an object influences the result as it occupies different amounts of the image.
- External light sources: In general it can be determined that the clearer the light edge is visible, the better the network estimates the pitch angle. However, we could not find any particular relation between oncoming cars or surrounding non-vehicular lights and predictions. At least, this change is not higher than the average standard deviation of predictions. However, it is pretty clear that too much light will deteriorate the results. An example image is given in Fig.9 (a).
- Slope: The training data contains almost inclination free road. Hence, stronger inclination of the road is a potential for errors. Please refer to Fig.9 (b).

## VI. CONCLUSION

In this paper we show that it is possible to estimate the headlight range and detect misaligned headlight calibration from monocular ADAS camera images in combination with DNNs. We implement a non-learning baseline using traditional computer vision techniques to compare our learning approach to. However, it proves to be difficult to handcraft a high-precision method that is robust against various environmental changes. The presented DNN approach clearly outperforms the baseline in terms of precision and robustness and additionally consumes less development time. Another advantage of the DNN is that a mono camera is sufficient to address the problem. We present a first working prototype that we plan to further investigate. Especially the recording of more diverse training data holds potential for further improvements. The advantage of our approach is that no additional sensors need to be integrated since we solely use the already built-in ADAS cameras.

## REFERENCES


[1] D. Eigen, C. Puhrsch, and R. Fergus, "Depth Map Prediction from a Single Image using a Multi-Scale Deep Network," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 2366–2374.

[2] L. Yang, "Comprehensive Visibility Indicator Algorithm for Adaptable Speed Limit Control in Intelligent Transportation Systems," PhD thesis, University of Guelph, Guelph, 03-2018.

[3] Ying Zhang, "3D Information Extraction Based on GPU," 2010.

[4] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka, "A stereo machine for video-rate dense depth mapping and its new applications," in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1996, pp. 196–202.

[5] M. Rezaei, M. Terauchi, and R. Klette, "Robust Vehicle Detection and Distance Estimation Under Challenging Lighting Conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2723–2743, 2015.

[6] P. F. Alcantarilla, L. M. Bergasa, P. Jiménez, I. Parra, D. F. Llorca, M. A. Sotelo, and S. S. Mayoral, "Automatic LightBeam Controller for driver assistance," *Machine Vision and Applications*, vol. 22, no. 5, pp. 819–835, 2011.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 770–778.

[9] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016.

[10] Alex 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.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 1–9.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2015.