LLNet: A Fusion Classification Network for Land Localization in Real-World Scenarios

Kun Chang and Li Yan *

School of Geodesy and Geomatics, Wuhan University, 129 Luoyu Road, Wuhan 430079, China; changkun@whu.edu.cn
* Correspondence: lyan@sgg.whu.edu.cn

Abstract: Lane localization is one of the core tasks in an autonomous driving system. It receives the visual information collected by the camera and the lane marks and road edges information outputted from the perception module and gives lane index for the subsequent decision module. Traditional rule-based lane localization methods using navigation maps can only be effective in regular road scenarios and have poor generalization ability. High-Definition Map (HD map) was originally thought to solve the lane localization problem, but due to the regulations of the relevant departments, HD map is currently not allowed to be used in autonomous driving systems. In addition, many multi-sensor fusion methods have been proposed to solve the lane localization problem. However, due to the extremely strict safety requirements of autonomous driving systems, these well-designed solutions make it difficult to meet the requirements in terms of robustness, efficiency, and stability. To solve these problems, we innovatively define the lane localization task as a classification problem. First, to better utilize the perceptual information outputted from the perceptual model, we design an image-generating method that projects the perceptual information onto a new image and ensures that our model can learn the perceptual features wisely. Second, to better fuse the perceptual and visual information, we propose a fusion structure deep learning neural network named LLNet to address the lane localization problem in an end-to-end manner. Finally, to ensure the generalization ability, robustness, and stability of LLNet, we conduct extensive comparison experiments on a large-scale real-world dataset, with a total mileage of over 400 km. The experiments show that our approach remarkably outperforms the deep learning classification baselines. In the discussion part of this paper, we give a comprehensive and detailed elaboration for the effectiveness of various designs in our LLNet. To our knowledge, LLNet is the first lane localization method based entirely on deep learning. LLNet is added to the self-driving suite for a kind of mass production vehicle that will be available in the summer of 2022, with an expected sales volume more than 300,000.

Keywords: lane localization; classification; deep learning

1. Introduction

Autonomous driving technologies in the last five years have not only drawn attention from both researchers and the public but have become the true driver-assistant in many scenarios, including turnpike and city expressway. For example, the Autopilot self-driving system of Tesla utilizes eight external cameras and 12 ultrasonic sensors, and a powerful onboard computer provides perception ability beyond human level. Tesla announced that in the first quarter of 2021, the accident rate of their Autopilot self-driving system was one accident for every 4.68 million miles driven. In contrast, the Insurance Institute for Highway Safety (IIHS) reported that the fatality rate per million miles traveled in 2019 was 1.11. The advantage of autonomous driving systems over humans is statistically obvious, but the application of this system is limited by legal and ethical practice. Constrained by the regulation department, the autonomous driving system is better known as an advanced driving assistance system (ADAS) [1–3] and employed in regular conditions such as the...
In turnpike, in which the component of the environment is relatively simple and the behavior of others is highly predictable. Lane localization plays a significant role in autonomous driving systems. In addition to the cruise control function, the routing planning and driving decision of vehicles, to a large extent, depend on the result of lane localization. This problem seems to be solved with the advancement of a High-Definition Map (HD map), through which the position of vehicles matches the lane information to achieve accurate localization. However, the HD map is not allowed to be the database for personal vehicle use in most countries. Therefore, lane localization problems need other solutions. Currently, with the development of deep learning, many companies and organizations employ finely designed DNN to simulate human perception ability. The high-level semantic features extracted from the vision information (in most cases, the videos or images accumulated from the cameras that are equipped in the vehicles) aid the onboard computer to decide the lane index. Despite the remarkable feature extraction ability of DNN, lane localization is still a difficult problem due to the following reasons:

- **Weak generalization ability in the real world.** Recently, many finely designed architectures of DNN have worked effectively in public databases, even achieving state-of-the-art performance. Some of them perform perfectly in the simulation system [4]. However, the real world is much more complex and complicated than the public database and simulation system. Given that the autonomous driving suite influences not only the drivers but also the walkers, the generalization ability of the architecture must be strong and robust. In the real world, many situations such as strong light, darkness, rain and moisture seriously affect the feature extraction ability of the algorithm. A less robust architecture may infer the wrong feature and then output the wrong lane index, which is likely to result in traffic accidents and endanger human life.

- **The limitation of on-board chips and computer.** Although some backbones in recent years have worked effectively or efficiently in large-scale image datasets, such as ImageNet, Coco or even real scenarios, noticeably, their performance mostly relies on the floating-point computing capability of Nvidia GPU (GTX series, RTX series and T4). The autonomous driving suite, being both equipment and a commodity, needs to have a balance between computational cost and effectiveness. In the first quarter of 2022, the most widely used chip for the autonomous driving system was the Snapdragon 8155. This Snapdragon processor includes eight Kry CPUs, a Qualcomm Adren 640 GPU and high-performance Hexagon 6 DSP, which has remarkable floating point computing capability as a portable chip but is still inferior to the PC-level counterparts. Furthermore, though the transformer framework shows considerable potential in vision tasks, it is not perfectly supported by the Snapdragon chips. The computing accumulation toolkit cannot be used on the transformer framework. Hence, the backbones selected have to be based on early frameworks rather than the transformer.

- **Expensive cost on labeled data.** In a real-world mass production scenario, the cost of labeled data is a huge problem. To train a robust network, a large amount of data with specific labels are fed into the net. In fact, an effective network in autonomous driving takes a hundred thousand or even millions of images as input. A specific task usually needs corresponding labeled images. In other words, it is less likely that a labeled image that meets the needs of all tasks exists. For example, the front vehicles detection and road edge detection are the priority while driving. The magnitude of these kinds of labeled data for the function is millions. In contrast, to address the lane localization problem, only ten thousand or a hundred thousand labeled images can be utilized. Therefore, to wisely use the output of other tasks is a helpful way to address the problem of lack of data.

To address the problems listed above, we propose a robust and effectively framework named LLNet. Our main contribution is as follows:

1. We view the lane localization problem as a classification task. To this end, we propose a light, effective, robust lane localization network (LLNet), whose generalized ability...
is considerably superior to the baseline DNNs. Aided by the dual-classifier heads, LLNet fully utilizes the information of both the images that are extracted from the vehicles equipped cameras and the total lane number that are loaded from the SDmap.

2. To increase the generalization capability of LLNet, we used the lane marks and road edges outputted from the perception model in the autonomous driving system to improve the performance of the classification network. To improve the training efficiency as well as to allow the classifier to better utilize the perception information, we generated an image of the perceptual data (As shown in Section 3.1). This image and the corresponding visual image became image pairs and were fed to the classification network simultaneously in a loose coupling manner to generate classification results. We assigned a small confidence weight to the perceptual information due to the instability of the output of the perception module.

3. To validate the generalized ability and effectiveness of LLNet, we carried out a comprehensive experiment on the data in the real-world scenario (Beijing and Guangdong province, China), which was collected and labeled by the profession data company. Due to commercial confidentiality requirements, only part of the test results are provided in this paper. The experiment results show that the proposed LLNet outweighs the baseline classifiers in precision, recall and generalization ability. We also conducted comparison experiments between different versions of LLNet to verify the rationality of the design of each part of LLNet. The experimental results show that the current LLNET design achieves an optimal balance between performance and computational consumption.

The lane localization algorithm proposed in the paper will be employed by a mass production vehicle with autonomous driving suite in the second quarter of 2022. The number of vehicles is expected to be 300,000. The algorithm may be improved in the final version and will be updated in the following time.

2. Related Work

Originally, the lane localization task was viewed as a simple subtask of precise localization of vehicles. In the driving process, the accurate position of the vehicles is supposed to be calculated with the high precision Inertial Navigation Systems (INS) [5,6], the global navigation satellite system (GNSS) [7–9], and the Inertial Measurement Unit (IMU) [10–12]. Christopher Rose, Jordan Britt et.al [13] fused the measurements from both camera and lidar, calculating lateral distance through two lane localization systems. The estimated results were utilized by the navigation filter to match a well-established waypoint-based map. The output can be provided in a GPS/INS system. Qingquan Li, Long Chen et al. [14] made use of the light, lidar and vision data to generate a fused real-time lane detection system. The fusion approach was constructed on feature-level rather pixel-level. Based on an optical selection strategy for the route prediction, the lane detection algorithm took the classification of potential drivable route as reference to selectively execute. Farhad Aghili, Chun-Yi Su et al. [15] combined the noise-adaptive Kalman filter and ICP registration algorithm to construct a six-degree-of-freedom relative navigation system. The output of the combination module was transformed by the laser scanner and IMU data. In addition, to improve the robustness of the system, the ICP alignment error estimation was calculated and the ICP point matching was checked by the fault detection logic. This navigation system was robust in the environment in which closed loop check can be performed. However, in real world application of autonomous driving, it was less likely that such a loop route exists. Aoki Takanose; Yuki Kitsukawa et al. [16] exploited the averaging effect by utilizing the time series data. The estimation result was further promoted by a specific module that minimized the GNSS multipath effect. E Soltanaghaei, A Prabhakara et al. [17] took advantage of the radar data of mmwave frequency to overcome the free space path loss problem. They depended on the signals from the tag at mmwave bands through the Van Atta Arrays.
However, due to the multi-path effects [18–20], which resulted from the reflection, and occlusion by buildings and other obstacles, the localization results were not stable enough to meet the requirement of autonomous driving. The information received by the sensor is often limited by the operating range and environmental conditions [21,22]. To address the problem of instability, Google [23] modeled the driving environment as a probabilistic map, in which the remittance values of each block represented its own gaussian distribution. The following Bayesian inference was able to reduce the uncertainty of the consistent angular reflectivity. Moreover, the offline SLAM was employed to align the multiple passes of the same environment. With the above steps, the driving environment was reconstructed with a high confidentiality. Furthermore, lane localization algorithms based on vision information were paid much attention, not only due to the low cost of cameras but the simulation of human drivers. Versaci M, Calcagno S et al. [24] proposed a fuzzy pre-processing based on geometric techniques with reduced computational load. K Zhao, Kang J et al. [25] presented a method combining Mask R-CNN with building boundary regularization to solve the problem of localizing all building polygons in the given satellite images. R Labayrade, J Douret et al. [26] designed three lane localization algorithms, each of which was based on different resolution of input data. The results of two low-level algorithms were fused to generate a reliability indicator. In parallel, the third algorithm based on the high-level data outputted a model by Kalman filtering. The prediction of the searching areas in the next image was given by the combination of the three algorithms. However, this method usually generated confused results when the number of lanes were over 3. Yan Jiang, Feng Gao et al. [27] designed a vision system based on the estimate and detect scheme to output the precise lane index. A perspective transformation was designed to output the estimation by scanning the row in the top view image. M Danielczuk, M Matl et al. [28] presented Synthetic Depth (SD) Mask R-CNN to segment unknown objects in depth images and has the potential to enhance robot skills in grasping and object tracking.

With the development of deep learning, the high-level semantic features extracted by DNNs gradually replaced the low-level geometric features such as conner, edge and simple texture. Noha Radwan, Abhinav Valada et al. proposed VLocNet++ [29] to analyze the inter-task relationship among semantic features learned by DNN, the regressing 6-DoF global pose, and odometry. VLocNet++ embedded the geometric and semantic knowledge of the world into the architecture in order to avoid the limitation of scene specific encoding constrains. A self-supervised warping step was added in the VLocNet++, by which intermediate network representations for sophisticated semantics was warped based on the relative motion. C Toft, E Stenborg et al. [30] proposed a score system to evaluate the similarity between the semantic features of the query image and scene. The consistency score was positively related to similarity, and the score was transmitted into a standard position pipeline to output the localization performance and then update the parameters of the system. Moreover, the labeled 3D point maps, combined with the semantically segmented images, can be used in the lane localization task. Erik Stenborg, Carl Toft et al. [30] trained a DNN to generate an image segmenter. The outputted map was then transformed into a localization result by a particle filter. Johannes L. Schönberger et al. [31] proposed a descriptor learning strategy by leveraging a generative model and studying semantic scene completion as an auxiliary task. In the work, the high-level 3D geometric and semantic information were encoded to difficult conditions, e.g., long time localization and missing observations condition. Based on the enhanced map data constructed from the OSM (Open Street Map), Philipp, Frank et al. [32] measured the relative position of the lane segment by using camera information to match. The initial pose guess was generated from the GNSS data; odometry was then updated and pose correction was performed by detected landmarks. Der-Hau Lee, Jinn-Liang Liu et al. [33] treated the lane localization problem as a segmentation task. They presented a DNN that is modified from the baseline segmentation network Unet and generated a simulation model by leveraging the PP algorithm with CNN. The model was used to output the quantity evaluation of the fusion network. Raja Muthalagu, Anudeepsekhar Bolimera et al. [34]
designed a joint framework to fuse the result of a lane mask proposal network and a lane key-point determination network. The aim of the framework was to output a key point prediction that described the left and right lane-markings of the vehicle lanes. Making full use of the semantic cues such as traffic lights and lane markings, Wei-Kang Tseng, Angela P. Schoellig et al. [35] calibrated the offset between the live GPS and semantic map frames. The GPS information and camera images for semantic cue detection were combined by an iterated extended Kalman Filter. Tong Qin, Yuxin Zheng et al. [36] proposed a framework that consist of mapping, on-cloud maintenance, and localization modules. They merged several localization results of surrounding vehicles and outputted the optimal estimation of the self-localization. Furthermore, due to the limitation of chip on board, lightweight DNN and accelerated inference were also paid much attention [37–43].

3. Method

Given a vision image extracted from the vehicle-equipped camera video $I_v$ and the road perception data outputted from the perception model $D_p$, the purpose of our proposed model is to wisely utilize the information from two sources. To this end, a fused model based on ResNet18 was designed to address the lane localization task as a classification problem. To better leverage the vision and perception information, a perception image generating step will be performed in the first place. Subsequently, the vision image and perception image are sent into the fused model simultaneously. A new loss function is designed to balance the effectiveness of the vision information and the perception information.

3.1. The Perception Image Generating Method

In the autonomous driving system, an original idea is that a highly complicated model, which collects all information derived from the various sensors equipped on the car, can be effective enough to output the features needed to control the vehicle. However, due to importance of safety for the autonomous driving system, a complicated fusion model for all tasks is not seen as a reasonable way because of the potential dangers of a high coupling mechanism. Once the model outputs the low-confidentiality feature, all downstream tasks such as threatened objects detection and road edge detection will output the instable or even absolutely wrong results, which may cause serious problems or threats for the autonomous driving system. The instable result outputted from the perception model is shown in Figure 1. Therefore, low coupling design is required because the extra redundancy brings enough safety for the system.

![Perception Result (a)](image1.png)  ![Perception Result (b)](image2.png)

**Figure 1.** The perception result outputted from the perception model. The blue line refers to the road edges; the yellow line refers to the middle line of the road; the white line refers to the lane mark. The left image is the correct classification output, whereas the right image is the wrong classification output. The reason why these lines are not absolute linear is that the perception model outputs several points instead of lines that can represent the lane mark and road edges. Since the vehicle is running, the result will be unstable once the perception model output the line.

Considering the lane mark and road edges data outputted from the perception model consist of several lines that label the lane mark and the road edges instead of an image, how to leverage the perception information and vision image information is a question that should be investigated. There are two ways that may address the problem: the high
coupling fusion and low coupling fusion. The high coupling fuse is shown in Figure 1. The fused image can be directly sent into the classification model. Theoretically, this high coupling fuse generates highly accurate results. However, this kind of fusion method reduces both precision and recall in practice (15% decline in precision, 22% decline in recall through our experiment). We suppose the reason for the decline is that the DNN for classification task pays more attention to the rectangle objects than the linear ones. In the lane localization task, it is the lane mark and road edges, rather than the surrounding objects, that assist in determining the lane index. Hence, a new fused image of low coupling design is proposed to overcome this dilemma. The fused method is shown in Figure 2.

![Vision image](image1.png) ![Perception image](image2.png)

**Figure 2.** The left image is the vision image extracted from the camera video; the right image is the corresponding perception image. The generating method is drawing the lane marks and road edges in a random image whose shape is the same as the original vision image.

The reason for drawing lines that represent the lane marks and road edges is that the advantage of the DNN is its ability to learn the high-level features, which may be extremely complex but consists of patterns. The surrounding environment is replaced by the random generated meaningless pixels. This step forces the DNN pay attention to the meaningful lines and output the classification result based on these lines.

### 3.2. The Architecture of the Fusion Model

In the computer vision field, many finely designed classifier networks are used to extract the high-level features of an image to address various tasks. However, due to the seriousness of the autonomous driving task, the classifier used in our model must be stable and robust with high generalization ability. After testing many classifier networks as our baseline, only the Transformer series and the ResNet series met the requirements of autonomous driving. Due to the limitations of the Snapdragon processor equipped in our vehicles, we choose the ResNet as our baseline model. As the lane marks and road edges, rather than the surrounding objects, are essential to the classification process, ResNet18 is superior to the ResNet50 and ResNet101, in which the lane marks that are tens of pixels wide cannot map into the final feature map because deep networks have more convolution and downsampling steps. Moreover, the FLOPS of ResNet18 is considerably lower than that of other two ResNets and thus require less inference time, which is crucial for the autonomous driving task.

The architecture of our proposed model is shown in Figures 3 and 4. The LLNet consists of two modules, a vision classification module and a perception classification module. The vision image $I_v$ and its perception counterpart $I_p$ are sent into the corresponding classification module, respectively. Additionally, the heads output the classification probability distribution. There are some differences in forward propagation between the training process and the inference process. In the training process, the output from each head in the vision classification module corresponds to its own label and calculates the loss by adding a step with a corresponding weight (as shown in Section 3.3). In the inference process, only a single head in each module outputs the result and merges the result from the perception module, predicting the final result based on the merged value.
The backbone in LLNet is a modified version of the ResNet18, the two layers of which (average pooling, 1000-d fully connected layer) are replaced by an adaptive average pooling layer and a fully connected layer whose dimension is based on the total lane (5 in our experiment because the 1–4 lane indexes cover 99.1% in all condition; 0 lane index represents inability to classify). The head in the vision classification module of LLNet refers to the SoftMax output that corresponds to different labels. A head corresponds to a label that counts the lane index from left to right, whereas another head corresponds to a label that counts the lane index in the opposite direction. The forward propagation in the training phase is shown in Equations (1)–(3)

\[ p_{\text{head} - 1} = f_{\text{head} - 1}(\theta_{Iv}) \]  
\[ p_{\text{head} - 2} = f_{\text{head} - 2}(\theta_{Iv}) \]
\[ p_{\text{head}−3} = f_{\text{head}−3}(\theta_I_p) \]

where \( p \) denotes the probability distribution outputted from a corresponding head (the SoftMax layer), \( f \) denotes mapping function that takes image \( I_v \) and \( I_p \) as input and gives the probability distribution as output, \( \theta \) denotes the parameter in the mapping function.

The forward propagation in the inference phase is shown in Equation (4)

\[ p_{\text{fuse}} = \alpha \cdot p_{\text{head}−1} + \beta \cdot p_{\text{head}−3} \]

where \( \alpha \) denotes the output weight of a vision classification module, we set 0.7 as default; \( \beta \) denotes the output weight of perception classification module, we set 0.3 as default.

The reason for using two labels from opposite directions is that through experiments, we found that the single head module based on the ResNet18 pays more attention to the left part of the image if only the label of left to right is taken as the input. In other words, the right side of the images, including the road edges and even the lane mark in the right side of the vehicle, is ignored by the DNN. Furthermore, the two head label contains more constraint and forces the network converge with more precision, compared with a single label. In the inference process, only a single head in the vision classification module is used to output the probability distribution. The purpose of this single output, rather than leveraging the dual heads outputs, is that when the dual head outputs different results, we cannot determine which result has a higher confidence. For example, assuming that the predicted index the vision classification heads output is lane 1 (left to right) and 2 (right to left), but the total lane is 4, these results cannot be distinguished because the same backbone with same parameters is used in the forward propagation process.

3.3. The Fusion Method and the Loss Function

As discussed above, the loss in the training phase and the validate (inference) phase is different. Since we viewed the lane localization problem as a classification task, the cross-entropy loss function was used as the base loss function in the proposed LLNet. The cross-entropy loss function is shown in Equation (5)

\[ L_{CE} = -\log q(c_i) \]

where \( q \) denotes the predicted probability distribution and \( c_i \) denotes the \( i \)-th class.

The loss function in the training phase is shown in Equations (6)–(9)

\[ L_{\text{head}−1} = L_{CE}(p_{\text{head}−1}, \text{label}−1) \]
\[ L_{\text{head}−2} = L_{CE}(p_{\text{head}−2}, \text{label}−2) \]
\[ L_{\text{head}−3} = L_{CE}(p_{\text{head}−3}, \text{label}−1) \]
\[ L_{\text{train}} = \alpha \cdot L_{\text{head}−1} + \beta \cdot L_{\text{head}−2} + \gamma \cdot L_{\text{head}−3} \]

where \( \alpha, \beta, \gamma \) denote the weight of the loss from head 1,2 and 3 respectively. We set them as 0.35, 0.35 and 0.3 as default.

The loss function in the validating (inference) phase is different from the training phase and is shown in Equation (10)

\[ L_{\text{val}} = 2 \cdot \alpha \cdot L_{\text{head}−1} + \gamma \cdot L_{\text{head}−3} \]

As is mentioned in Section 3.2, the forward propagation in the validating or inference phase is based on the single output of the vision classification module. Due to the equal confidence of the output from both heads, the weight of loss of head-1 increases from the 0.35 to 0.7. The reason we set the weight of loss from the perception image remarkably inferior to that from vision image is that perception data is derived from the perception DNN rather than the original data. Although the perception model is trained on data
whose magnitude is up to millions because of its importance and superiority, the data outputted from perception model is not as reliable as the data from the video camera. The vision images extracted from the video camera provide original information with only a few mistakes in some extreme conditions. This distorted visual information due to camera overexposure or motion blurring can be easily identified by PNSR and other indicators. Conversely, errors, drifts, and unstable values in the output of the perceptual model are difficult to determine or evaluate. That is, it is difficult to have an index that can reasonably assess the confidence level of road edges and lane lines outputted from the perception model. Therefore, only a relatively low weight can be given to the classification results of the perceptual model output in the LLNet.

4. Experiments and Results
4.1. Testset
4.1.1. Introduction
Since our lane line localization algorithm is to be set up in a mass production vehicle by mid-2022, our comparison experiments are tested in real-world scenarios on a very large scale. The entire dataset includes 11 long-distance routes in Beijing and Guangdong Province, China, with a total mileage of over 400 km. To deeply evaluate the generalization capability of our model, the test set covers a variety of scenarios, including highways, urban expressways, ordinary urban lanes, traffic lights, traffic circles, and a series of other scenarios with very large differences. Due to the difference in urban design and construction style, the test roads in Beijing are very different from those in Guangzhou. The cameras used in the real autonomous driving systems equipped in the vehicles output 30 frames images per second, and only 1 frame is extracted as the model input in this comparison experiment. Due to the low sampling rate in the test, the output of the perception model seems to be incomplete in terms of continuity, which also challenges the stability and robustness of the classification output obtained by our model in the case of poor sampling rate. The input data of LLNet in a real scene would reach 15 frames per second. However, in the comparison experiments in this paper, we set a lower sampling rate because we are more interested in testing the generalization ability of the model in different environments and situations. The total number of datasets is 94,112, all of which are downsampled from 2.7 million images by 30:1. The training set has 54,077 images and the test set has 40,315 images. Unlike the ImageNet and COCO datasets, the training and test sets for the classification task in autonomous driving are strictly ordered by time series. In addition, the selection of the validation set cannot rely on random sampling but contains data from the entire line so as to ensure that the generalization ability of the model can be verified through the validating loss. The setup of the validation set and the early stop mechanism in the lane localization task are noticeably complicated and are not presented in detail in this paper. The data distribution is shown in Figure 5.

As shown in Figure 5, the number of first and second lane index exceeds 80% of the overall data, which is also consistent with the real-world scenario: most roads are two lanes in one direction, so the number of first and second lanes far exceeds the rest of the classes. Since deep learning relies heavily on the amount of data, this uneven data distribution also poses some pressure for the effectiveness of the network.

Due to the requirement of commercial confidentiality agreement, we can only present part of the test results.
4.1.2. Pre-Processing Step

Our dataset comes from the vehicle-equipped camera. The raw output of the camera is \(3840 \times 1920\) pixels. Despite dense and precise semantic information due to the high resolution camera, feeding the high-resolution images directly into the network would greatly increase the training and inference time expenditure, which does not meet the requirement of real-time processing. Therefore, these high-resolution images need to be pre-processed before being conveyed into the network.

As shown in Figure 6, the visual image output from the camera is first resized to \(480 \times 240\), and then the crop step is performed in a height ranging from 50 to 160. The final output image size is \(480 \times 110\). The reason of resizing to \(480 \times 240\) is that a lower compression ratio would make the input image lose crucial semantic and texture features. The thin lane marks and road edges would be compressed into only a few pixels wide. Additionally, due to the convolution and downsampling step, the lane marks less than 16 pixels wide would not be reflected into the feature map. In addition, the reason for the crop operation from 50 to 160 is that the area with a height between 0 and 50 is filled with mainly car covers, which do not contain useful classification features for lane detection. Moreover, the area with heights between 110 and 240 are mainly sky and also do not contain valid visual information. As a result, in order to accelerate the training and inferring process, visual images with a shape \(480 \times 110\) are obtained by the above pre-processing steps. The generation of the perceptual image has been described in detail in Section 3.1. After the pre-processing step, the pair of visual images and the corresponding perceptual images are generated.

Since one-way less than or equal to four lanes can already cover the majority of real-world scenarios, all lane indexes greater than 4 are given a label 0, which means that they cannot be discriminated during training and testing. There are 103 categories of lane indexes in real engineering scenarios which are intended to cope with various corner cases. A total of 103 categories are not explored in depth in this paper.
4.2. Setup

4.2.1. The Evaluation Metrics

Choosing reasonable and effective metrics for evaluation is a necessary task in the fields of Computer Vision (CV), natural language processing (NLP), and information retrieval (IR). In classification tasks, the most commonly used metrics are accuracy, precision, and recall. Accuracy refers to the ratio of the number of samples correctly classified by the classifier to the total number of samples for a given test dataset; that is, the accuracy of the test dataset when the loss function is a 0–1 loss. Precision refers to the fraction of relevant instances retrieved; Recall refers to the fraction of relevant instances retrieved. The definition formulas of accuracy, precision and recall are shown in Equations (11)–(13)

\[
\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}
\]

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{12}
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{13}
\]

where \(TP\) refers to the number of true positives, \(FP\) refers to the number of false positives, \(TN\) refers to the number of true negatives, and \(FN\) refers to the number of false negatives.

Accuracy is the evaluation metric that best matches human intuition. However, in an autonomous driving task, accuracy becomes a metric that has little practical value, because the undeterminable true-negative samples are little useful information for the decision module of autonomous driving system. Precision and Recall are much more valuable metrics. Furthermore, precision is more important than recall rate, because autonomous driving system places great emphasis on safety. In details, true-positive results are very important for the decision module to output correct judgments, while false-positive results can be misjudged by the decision module and therefore seriously jeopardize the effectiveness of the autonomous driving system.

4.2.2. Setting

During the training process, pre-processed visual and perceptual image pairs were fed into LLNet simultaneously. The training and testing part of the experiments were carried out on a deep learning server with an Nvidia V100 GPU and 64 GB memory. Batch size was set to 256. The optimizer is an Adam optimizer. Data pre-processing was carried out by OpenCV. The version of deep learning framework is pytorch 1.10.1.

4.2.3. Training Process

Because our LLNet is constructed on the basis of ResNet, LLNet does not have many hyperparameters. The most critical hyperparameter is the learning rate, which is set to 0.01 in our training process. Given that a small batch size (less than 64) will greatly slow down the training process (the number of lane detection images is considerable),
we kept a relatively large batch size during the training process to ensure the training speed. Throughout our experiments, when setting the batch size 64, 128, 256, and 512, there was basically no difference in the final converge. This is because when the batch size is larger than 64, each batch basically approximates the features of the whole dataset, and there is no significant fluctuation in the parameter update of BN layer. In contrast, when the batch size is set to 16, 8, 4, or 2, the feature distribution of different batches differs significantly, resulting in significant fluctuations in the BN layer and thus adversely affecting the convergence. When the GPU memory is sufficient, we recommend using a large batch size.

The training process of LLNet is pretty fast, and the convergence is usually reached within 4 epochs. As shown in Figure 7, the loss decreases rapidly with the increase in epochs, while the accuracy considerably improves as the training process proceeds. However, overfitting is a problem that needs attention. In our experiments, overfitting is not severe due to the fact that the amount of the training set is over 270 thousand images and the network can learn the most important features of the train dataset. When the training set is relatively small, the overfitting problem is more severe. It is usually necessary to design models with better feature extraction ability to mitigate the overfitting problem.

The results from Tables 1–7 show that our LLNet is significantly superior to the baseline classification network in terms of precision and recall metrics for all lane classes, which proves the effectiveness of our fuse strategy. It is worth noting that the classification results for all networks are significantly better in lanes 1 and 2 than in lanes 3 and 4. The reason for this contrast is the uneven distribution of data. In real-world scenarios, the number of roads with two lanes in one direction is much higher than the number of roads with four lanes in one direction, so the number of first and second lanes is much higher than the number of third lanes and third and fourth lanes. Since the generalization performance of the deep learning model is highly dependent on the amount of data, the network learns more features of the first two lanes, so the generalization performance will be better than that for lane 3 and 4.

| Lane Index | Precision (%) | Recall (%) |
|------------|---------------|------------|
| Lane-1     | 55.3          | 51.2       |
| Lane-2     | 41.9          | 49.2       |
| Lane-3     | 22.8          | 32.7       |
| Lane-4     | 32.4          | 28.0       |

Figure 7. Loss and Accuracy fluctuation in different Epoch.

4.3. Experiment Result

4.3.1. Comparison with Baselines

In order to comprehensively evaluate the performance of LLNet, we selected the most classical classification network baselines in deep learning as the comparison networks, including AlexNet [44], Vgg16 [45], Resnet18 [46], and Resnet50 [46]. The comparison results are shown below.

The results from Tables 1–7 show that our LLNet is significantly superior to the baseline classification network in terms of precision and recall metrics for all lane classes, which proves the effectiveness of our fuse strategy. It is worth noting that the classification results for all networks are significantly better in lanes 1 and 2 than in lanes 3 and 4. The reason for this contrast is the uneven distribution of data. In real-world scenarios, the number of roads with two lanes in one direction is much higher than the number of roads with four lanes in one direction, so the number of first and second lanes is much higher than the number of third lanes and third and fourth lanes. Since the generalization performance of the deep learning model is highly dependent on the amount of data, the network learns more features of the first two lanes, so the generalization performance will be better than that for lane 3 and 4.
Table 2. Classification Result of Vgg16 (vision image).

| Lane Index | Precision (%) | Recall (%) |
|------------|---------------|------------|
| Lane-1     | 63.9          | 66.7       |
| Lane-2     | 67.0          | 61.1       |
| Lane-3     | 43.7          | 44.1       |
| Lane-4     | 41.3          | 37.4       |

Table 3. Classification Result of ResNet18 (vision image).

| Lane Index | Precision (%) | Recall (%) |
|------------|---------------|------------|
| Lane-1     | 84.2          | 87.3       |
| Lane-2     | 78.4          | 79.3       |
| Lane-3     | 51.5          | 67.0       |
| Lane-4     | 45.9          | 44.1       |

Table 4. Classification Result of ResNet18 (perception image).

| Lane Index | Precision (%) | Recall (%) |
|------------|---------------|------------|
| Lane-1     | 68.8          | 61.4       |
| Lane-2     | 63.0          | 50.8       |
| Lane-3     | 44.6          | 50.5       |
| Lane-4     | 41.5          | 32.3       |

Table 5. Classification Result of ResNet50 (vision image).

| Lane Index | Precision (%) | Recall (%) |
|------------|---------------|------------|
| Lane-1     | 71.0          | 69.3       |
| Lane-2     | 66.1          | 73.8       |
| Lane-3     | 44.1          | 59.9       |
| Lane-4     | 42.8          | 43.3       |

Table 6. Classification Result of ResNet50 (perception image).

| Lane Index | Precision (%) | Recall (%) |
|------------|---------------|------------|
| Lane-1     | 41.2          | 30.0       |
| Lane-2     | 29.8          | 22.1       |
| Lane-3     | 24.4          | 39.2       |
| Lane-4     | 36.7          | 35.5       |

Table 7. Classification Result of LLNet (vision and perception image).

| Lane Index | Precision (%) | Recall (%) |
|------------|---------------|------------|
| Lane-1     | 95.6          | 94.8       |
| Lane-2     | 92.2          | 90.9       |
| Lane-3     | 70.9          | 69.3       |
| Lane-4     | 66.1          | 66.6       |

In addition, the comparison from Tables 3–6 shows that ResNet18 performs significantly better than ResNet50. This is because the shallow layer of the Resnet network extracts geometric features such as straight lines, curves, and angles, and the deeper layer focuses more on more complex semantic features. The most important lane marks and road edges for lane localization are lost in the convolution and downsampling operations and cannot be mapped to the deep feature map. Although the high-level semantic features of surrounding objects are retained, they do not help much for lane localization. Thus, too deep networks will instead hinder the classification performance in a lane localization task, which is opposed to the traditional object classification task.
Moreover, as shown in the comparison between Tables 3 and 4, and between Tables 5 and 6, using perceptual data alone for classification brings a significant degradation of performance. In other words, when using ResNet series as the classification network for lane localization problem, the visual information is more reliable than the perceptual information. This is because the visual information is less likely to have incorrect values. In contrast, the perceptual information is the result of the output of the perceptual model. Even though the perceptual model is trained on millions of images, it is possible to output wrong results in some cases, which can further provide wrong information to the classification model and thus make the process unstable. This is why we only set a low weight to the perceptual data in LLNet (Equations (4), (9) and (10)).

4.3.2. Comparison with Different Setting of LLNet

To further demonstrate the effectiveness of the fusion approach we designed, we compared LLNet in different settings, which include:

- LLNet-Original: the original LLNet
- LLNet-Vision: delete the perception module
- LLNet-single-vision-head: delete the dual heads in vision module
- LLNet-dual-perception-head: make the perception dual heads, which is same as the vision module

The comparison results are shown below.

As can be seen from Table 8, the original LLNet design achieves excellent classification performance in most cases. In detail, there is a significant decrease in precision and recall metrics in the LLNet-Vision. This is because the perceptual data is the result of the output of the perceptual model, which is trained on millions of data and therefore learns more generalized features. Comparison of the results of the visual and fusion models also justifies the data fusion performed by LLNet. In addition, the visual and perceptual fusion model with a single head does not differ significantly from the original network in terms of precision, but a remarkable decrease in the recall is obvious. The reason is that the lane index is calculated from left to right in the single-head labels, so the network pays more attention to the left part of the image and ignores the right part at the same time. The double-head design in LLNet is to allow the network to focus on both sides of the image. Moreover, the perceptual module version with two heads achieves a similar performance to the original LLNet, with a difference of 0.3 to 2.5. Since the improvement is not obvious, the original design is adequate for the lane localization task. It can also be seen that designing a dual head setting in the perception module does not provide a significant improvement. The reason is that most of the pixels in the perceptual image are randomly generated, and since the network does not learn effective features, it can only focus on the lane lines and road edges of the entire image, thus not ignoring the right part of the image just as the vision module.

Table 8. Classification Result of different versions of LLNet.

| Lane Index | Precision (%) | Recall (%) |
|------------|---------------|------------|
| Lane-1     | 95.6/89.1/91.7/95.1 | 94.8/91.9/79.3/95.0 |
| Lane-2     | 92.2/84.9/90.0/92.5 | 90.9/88.4/77.1/90.2 |
| Lane-3     | 70.9/63.1/69.8/70.4 | 69.3/66.4/50.8/69.7 |
| Lane-4     | 66.1/58.8/63.8/67.0 | 66.6/52.7/41.4/69.1 |

LLNet-Original/LLNet-Vision/LLNet-single-vision-head/LLNet-dual-perception-head.

5. Discussion

Lane positioning has been treated as a pure positioning problem. One common idea to address this problem is the joint optimization of multi-sensors such as GPS and IMU. Another idea is to use manually designed features to extract the lane marks, followed by rule-based localization. However, with the development of deep learning, DNNs are able to not only learn the high-level semantic features that are incomprehensible to humans
but also build end-to-end solutions. In addition, the generalization capability of DNN-based lane localization algorithms can be continuously promoted due to the accumulation of data. Therefore, we creatively define the lane localization problem as a classification problem in the deep learning field. To our knowledge, there never been a lane localization algorithm based entirely on deep learning before LLNet. However, directly applying the deep learning classification baselines does not meet the requirement of autonomous driving. This is because lane localization classification tasks are different from typical visual classification tasks. The reasons are as follows:

First, common data augmentation methods cannot be used because the images during the inference process do not have the same changes as the data augmentation, as shown in Figure 8. Additionally, data augmentation methods such as rotation within a small angle (Figure 9) can be used in the lane localization task, because the slight rotation of camera does happen as the auto-driving suite is continuously working. The key point of data augmentation is that the augmentation condition must happen in real-world applications.

Second, the core features that lane localization relies on are lane marks and road edges. The high-level semantic features extracted by the classification baselines are helpful but not the most important for this lane localization task. This is a significant difference from the traditional classification task. As can be seen from Figures 10 and 11, ResNet outperforms the other baselines (AlexNet and VGG). However, unlike the traditional classification and object detection tasks, ResNet18 is superior to ResNet50. This is because the lane marks, which are important for the lane localization task, are not mapped to the final feature map due to downsampling and convolution operations in the deeper network.
The data augmentation methods that can be used in the lane localization task: Rotation within a small angle (in our setting less than 10 degree in clockwise and counterclockwise) such rotation commonly encountered in real world application.

Figure 9.

The requirement for precision and performance of the baselines and LLNet.

Figure 10.

The requirement for recall and performance of the baselines and LLNet.

Figure 11.

The visual images downsampled by the camera possess remarkable stability and robustness, but as shown in Table 9, the performance of the classification network relying
on visual information alone cannot meet the precision requirement of autonomous driving (90% precision). Therefore, we need additional information to improve the performance of the lane localization algorithm. In an autonomous driving system, the perceptual information is the output from the perceptual model. Due to the importance of the perception module, the perception model is trained on millions of labeled images, so the perception outputs usually have high-level semantic features. The road edges and lane routes from perception model can be used for the lane localization module. As also shown in Table 9, the precision metric can meet the requirement once LLNet utilizes the perception and vision information with a single classifier head.

**Table 9.** The comparison of ResNet18 between vision information only and the fusion input (LLNet with one head in vision module).

| Lane Index | Precision (%) | Recall (%) |
|------------|---------------|------------|
| Lane-1     | +7.5          | −8.0       |
| Lane-2     | +11.6         | −2.2       |
| Lane-3     | +18.3         | −17.0      |
| Lane-4     | +17.9         | −2.7       |

However, it should be noted that adding perceptual information significantly reduces the recall metric of the model, because the output of the perceptual model is unstable. Moreover, adding perceptual information actually places more constraints on the classification process, and a large number of scenes that can be judged by a single visual image would be classified as undecidable with the limitation of perceptual information. In general, adding perceptual information can improve precision but reduce recall at the same time.

In order to utilize both visual and perceptual information, a smarter fusion model is necessary. In addition, due to the requirement for real-time lane localization, loosely coupling models are more preferable than tightly coupling models. Therefore, how to loosely couple the visual and perceptual information is a question that should be addressed. The LLNet proposed in this paper makes use of visual and perceptual information for classification separately, followed by a weighted output. We give a relatively low weight to the perceptual information because the perceptual information is not as stable and robust as vision information. What is more, to address the decline due to adding the perception information, we design a dual-head architecture.

As shown in Table 10, the precision and recall of the dual-head output classification results is significantly higher than that of the single-head version, because the single-labeled lanes are computed from left to right, so the network ignores the information on the right side of the image. The dual-head architecture of LLNet forces the network to extract classification features from both directions equally.

**Table 10.** The comparison between single head ResNet18 and dual-head LLNet (vision image only).

| Lane Index | Precision (%) | Recall (%) |
|------------|---------------|------------|
| Lane-1     | +4.9          | +4.6       |
| Lane-2     | +6.5          | +9.1       |
| Lane-3     | +11.6         | −0.6       |
| Lane-4     | +12.9         | +8.6       |

To our knowledge, LLNet is the first lane localization network based entirely on deep learning and achieves the satisfying metrics in the first and second lanes as required by autonomous driving systems. However, our study still has some shortcomings.

1. The precision and recall metric in third and fourth lanes cannot meet the requirements, which is caused by the insufficient data in the third and fourth lanes. However, in the real world, the three-lane and four-lane in one direction is not a common scenario. Moreover, it is difficult to collect data exclusively for the third and fourth lanes because
the labeled data is very expensive. Furthermore, the lane localization task is different from the traditional classification task, and common data enhancement methods cannot be used. Therefore, how to achieve large-scale pre-labeling or suitable data augmentation is a critical problem which is necessary to be solved.

2. The vehicle driving process is a continuous process. Intuitively, the temporal sequence of visual and perceptual information is highly useful but currently not well utilized. This information could theoretically further improve the accuracy of lane localization in the future.

6. Conclusions

In this paper, we propose a fusion deep learning neural network named LLNet to solve the lane line localization problem in an end-to-end manner. Innovatively, we define the lane detection task as a classification problem. To make full use of the lane marks and road edges information outputted from the perception model, as well as the vision information from the camera, we designed a loosely coupling fusion network to perform the fusion of the visual classification results and perceptual classification result. In LLNet, the visual and perceptual images are passed through the classification model separately, and the probability distribution is outputted and then the weighted sum is calculated. To maintain stability and robustness and to improve the generalization performance, we use ResNet18 as the backbone. In addition, to overcome the problem that the classification model focuses more on the left side of the image and ignores the right side, we designed a multi-head mechanism in the vision module to force the network to focus equally on the classification features from the left and right sides. In the comparison experiments on a large-scale dataset of 11 roads with a total mileage of over 400 km in Beijing and Guangdong Province, the LLNet significantly outperforms the current baseline classification network in all metrics. Moreover, we conducted a comparison of different versions of LLNet to justify the effectiveness of the architecture.

Currently, our method has already met the requirement of precision and recall metrics in the first and second lane scenarios. However, due to the insufficient number of samples in the trainset, the metrics of the third and fourth lanes is relatively low. Semi-supervised learning methods may be effective in this task, and we take it as the future work.

Author Contributions: K.C. conceived and designed the experiment; K.C. and L.Y. provided the methodology; K.C. performed the experiment and analyzed the data; K.C. prepared the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key Research and Development Project of China: 2020YFD1100200; The Science and Technology Major Project of Hubei Province under Grant: 2021AAA010.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: All authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

| Abbreviation | Description                        |
|--------------|------------------------------------|
| HD map       | High-Definition Map                |
| IIHS         | Insurance Institute for Highway Safety |
| ADAS         | Advanced driving assistance system |
| LLNet        | Lane localization network          |
| INS          | Inertial Navigation Systems        |
| GNSS         | Global navigation satellite system |
| IMU          | Inertial Measurement Unit          |
| SD           | Synthetic Depth                    |
| OSM          | Open Street Map                    |
References

1. Luo, H.; Wang, C.; Wen, C.; Cai, Z.; Chen, Z.; Wang, H.; Yu, Y.; Li, J. Patch-based semantic labeling of road scene using colorized mobile LiDAR point clouds. *IEEE Trans. Intell. Transp. Syst.* **2015**, *17*, 1286–1297.

2. Tan, M.; Wang, B.; Wu, Z.; Wang, J.; Pan, G. Weakly supervised metric learning for traffic sign recognition in a LIDAR-equipped vehicle. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1415–1427.

3. Hata, A.Y.; Wolf, D.F. Feature detection for vehicle localization in urban environments using a multilayer LIDAR. *IEEE Trans. Intell. Transp. Syst.* **2015**, *17*, 420–429.

4. Mur-Artal, R.; Tardós, J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]

5. Dewangkan, D.K.; Sahu, S.P. Lane Detection for Intelligent Vehicle System Using Image Processing Techniques. In *Data Science*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 329–348.

6. Liu, D.; Cui, Y.; Guo, X.; Ding, W.; Yang, B.; Chen, Y. Visual localization for autonomous driving: Mapping the accurate location in the city maze. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 3170–3177.

7. Xia, Z.; Booij, O.; Manfredi, M.; Kooij, J.F.P. Cross-View Matching for Vehicle Localization by Learning Geographically Local Representations. *IEEE Robot. Autom. Lett.* **2021**, *6*, 5921–5928.

8. Torkayesh, A.E.; Zolfani, S.H.; Kahvand, M.; Khazaelpour, P. Landfill location selection for healthcare waste of urban areas using hybrid BWM-grey MARCOS model based on GIS. *Sustain. Cities Soc.* **2021**, *67*, 102712.

9. Ragothaman, S.; Maaref, M.; Kassas, Z.M. Autonomous ground vehicle path planning in urban environments using GNSS and cellular signals reliability maps: Models and algorithms. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 1562–1580.

10. Paden, B.; Cáple, M.; Yong, S.Z.; Yershov, D.; Frazzoli, E. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Trans. Intell. Veh.* **2016**, *1*, 33–55.

11. Li, L.; Yang, M.; Guo, L.; Wang, C.; Wang, B. Hierarchical neighborhood based precise localization for intelligent vehicles in urban environments. *IEEE Trans. Intell. Veh.* **2016**, *1*, 220–229. [CrossRef]

12. Takeyama, K.; Machida, T.; Kojima, Y.; Kubo, N. Improvement of dead reckoning in urban areas through integration of low-cost multisensors. *IEEE Trans. Intell. Veh.* **2017**, *2*, 278–287.

13. Rose, C.; Britt, J.; Allen, J.; Bevly, D. An integrated vehicle navigation system utilizing lane-detection and lateral position estimation systems in difficult environments for GPS. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2615–2629. [CrossRef]

14. Li, Q.; Chen, L.; Li, M.; Shaw, S.-L.; Nüchter, A. A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios. *IEEE Trans. Veh. Technol.* **2015**, *63*, 540–555. [CrossRef]

15. Aghilj, F.; Su, C.-Y. Robust relative navigation by integration of ICP and adaptive Kalman filter using laser scanner and IMU. *IEEE/ASME Trans. Mechatron.* **2016**, *21*, 2015–2026. [CrossRef]

16. Takanose, A.; Kitsukawa, Y.; Meguro, J.; Takeuchi, E.; Carballo, A.; Takeda, K. Eagleye: A Lane-Level Localization Using Low-Cost GNSS/IMU. In Proceedings of the 2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops), Nagoya, Japan, 11–17 July 2021; pp. 319–326.

17. Soltanaghaei, E.; Prabhakara, A.; Balanuta, A.; Anderson, M.; Rabaey, J.M.; Kumar, S.; Rowe, A. Millimetre: mmWave retro-reflective tags for accurate, long range localization. In Proceedings of the 27th Annual International Conference on Mobile Computing and Networking, New Orleans, LA, USA, 25–29 October 2021; pp. 69–82.

18. Choi, W.-S.; Kim, Y.-S.; Oh, S.-Y.; Lee, J. Fast iterative closest point framework for 3D LIDAR data in intelligent vehicle. In Proceedings of the 2012 IEEE Intelligent Vehicles Symposium, Madrid, Spain, 3–7 June 2012; pp. 1029–1034.

19. Zhang, J.; Singh, S. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 2174–2181.

20. Rohde, J.; Völz, B.; Mielenz, H.; Zöllner, J.M. Precise vehicle localization in dense urban environments. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 853–858. [CrossRef]

21. Ghassemian, H. A review of remote sensing image fusion methods. *Inf. Fusion* **2016**, *32*, 75–89. [CrossRef]

22. Witharana, C.; Bhuiyan, M.A.E.; Liljedahl, A.K.; Kanevski, M.; Epstein, H.E.; Jones, B.M.; Daanen, R.; Griffin, C.G.; Kent, K.; Jones, M.K.W. Understanding the synergies of deep learning and data fusion of multispectral and panchromatic high resolution commercial satellite imagery for automated ice-wedge polygon detection. *ISPRS J. Photogramm. Remote Sens.* **2020**, *170*, 174–191. [CrossRef]

23. Levinson, J.; Thrun, S. Robust vehicle localization in urban environments using probabilistic maps. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 4372–4378. [CrossRef]

24. Versaci, M.; Calcagno, S.; Morabito, F.C. *Image Contrast Enhancement by Distances among Points in Fuzzy Hyper-Cubes*; Springer: Cham, Switzerland, 2015; pp. 494–505.
25. Zhao, K.; Kang, J.; Jung, J.; Sohn, G. Building Extraction from Satellite Images Using Mask R-CNN with Building Boundary Regularization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018.

26. Labayrade, R.; Douret, J.; Laneurit, J.; Chapuis, R. A reliable and robust lane detection system based on the parallel use of three algorithms for driving safety assistance. *IEEE Trans. Inf. Syst.* 2006, 89, 2092–2100. [CrossRef]

27. Jiang, Y.; Gao, F.; Xu, G. Computer vision-based multiple-lane detection on straight road and in a curve. In Proceedings of the 2010 International Conference on Image Analysis and Signal Processing, Zhejiang, China, 9–11 April 2010; pp. 114–117.

28. Danielczuk, M.; Matl, M.; Gupta, S.; Li, A.; Goldberg, K. Segmenting Unknown 3D Objects from Real Depth Images using Mask R-CNN Trained on Synthetic Data. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019.

29. Radwan, N.; Valada, A.; Burgard, W. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robot. Autom. Lett.* 2018, 3, 4407–4414. [CrossRef]

30. Toft, C.; Stenberg, E.; Hammarstrand, L.; Brynke, L.; Pollefeys, M.; Sattler, T.; Kahl, F. Semantic match consistency for long-term visual localization. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 383–399.

31. Schönberger, J.L.; Pollefeys, M.; Geiger, A.; Sattler, T. Semantic visual localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6896–6906.

32. Philipp, F. Multiple Camera Road Perception and Lane Level Localization in Urban Areas. Ph.D. Thesis, Freie Universität Berlin, Berlin, Germany, 2021.

33. Lee, D.-H.; Liu, J.-L. End-to-End Deep Learning of Lane Detection and Path Prediction for Real-Time Autonomous Driving. *arXiv* 2021, arXiv:2102.04738.

34. Muthalagu, R.; Bolimera, A.; Kalaichelvi, V. Vehicle lane markings segmentation and keypoint determination using deep convolutional neural networks. *Multimed. Tools Appl.* 2021, 80, 11201–11215. [CrossRef]

35. Tseng, W.-K.; Schoellig, A.P.; Barfoot, T.D. Self-Calibration of the Offset Between GPS and Semantic Map Frames for Robust Localization. In Proceedings of the 2021 18th Conference on Robots and Vision (CRV), Burnaby, BC, Canada, 26–28 May 2021.

36. Qin, T.; Zheng, Y.; Chen, T.; Chen, Y.; Su, Q. RoadMap: A Light-Weight Semantic Map for Visual Localization towards Autonomous Driving. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi’an, China, 30 May–5 June 2021.

37. Garnett, N.; Cohen, R.; Pe’er, T.; Lahav, R.; Levi, D. 3d-lanenet: End-to-end 3d multiple lane detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 2921–2930.

38. Kim, W.; Kang, C.M.; Son, Y.S.; Lee, S.-H.; Chung, C.C. Vehicle path prediction using yaw acceleration for adaptive cruise control. *IEEE Trans. Intell. Transp. Syst.* 2018, 19, 3818–3829. [CrossRef]

39. Lin, C.-F.; Ulsoy, A.G.; LeBlanc, D.J. Vehicle dynamics and external disturbance estimation for vehicle path prediction. *IEEE Trans. Control. Syst. Technol.* 2000, 8, 508–518.

40. Cudrano, P.; Mentasti, S.; Matteucci, M.; Bersani, M.; Arrigoni, S.; Cheli, F. Advances in centerline estimation for autonomous lateral control. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 1415–1422.

41. Li, X.; Li, J.; Hu, X.; Yang, J. Line-cnn: End-to-end traffic line detection with line proposal unit. *IEEE Trans. Intell. Transp. Syst.* 2019, 21, 248–258. [CrossRef]

42. Zou, Q.; Jiang, H.; Dai, Q.; Yue, Y.; Chen, L.; Wang, Q. Robust lane detection from continuous driving scenes using deep neural networks. *IEEE Trans. Veh. Technol.* 2019, 69, 41–54. [CrossRef]

43. Lu, P.; Cui, C.; Xu, S.; Peng, H.; Wang, F. SUPER: A novel lane detection system. *IEEE Trans. Intell. Veh.* 2021, 6, 583–593. [CrossRef]

44. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* 2017, 60, 84–90. [CrossRef]

45. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* 2014, arXiv:1409.1556.

46. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.