Semantic Bird's-Eye View Road Line Mapping

Matteo Bellusci*, Paolo Cudrano*, Simone Mentasti*, Riccardo Erminio Filippo Cortelazzo†, and Matteo Matteucci*

***Abstract*— The development of Autonomous Vehicles (AVs) today requires precise and reliable detection of road line markings. Indeed, recognizing road line markings from camera images acquired by the vehicle plays a crucial role in ensuring its safe navigation and improving its driving performance. Road line detection is of key importance in real-time scenarios for navigation purposes, as well as offline for the generation of HD maps. In recent years, deep neural networks have proven effective in performing this task. In particular, Convolutional Neural Networks (CNNs) have helped develop multiple Advanced Driver Assistance Systems (ADAS), now fully integrated into common commercial vehicles. This paper presents a novel CNN-based pipeline for recognizing road line markings from front-view camera images in an online setup, and it shows how these detections can be aggregated offline into aerial-like maps as a first step toward the creation of HD maps. The proposed architecture comprises a multi-decoder to accurately classify image pixels representing different classes of road line markings, as well as those related to the drivable area. The mapping system then projects the extracted road line points into the Bird's-Eye View (BEV) space and integrates the extracted information with accurate localization measurements for georeferencing. Experimental evaluations on real-world data, including data acquired with instrumented vehicles, reveal the effectiveness of the proposed pipeline in both frame-by-frame detection and integrated mapping quality.**

I. INTRODUCTION

Technologies based on neural networks are increasingly being deployed in the automotive sector to solve a variety of problems, from improving the capabilities of Advanced Driver Assistance Systems (ADAS) to perception and control in Autonomous Vehicles (AVs) [1], [2]. Neural networks are trained to interpret the data acquired by the vehicle sensors and provide useful outputs to assist the human driver or the control system. A widely studied problem is road line detection. This task has traditionally been approached through procedural analysis of vehicle camera images, processed using traditional computer vision algorithms [3]. However, recent advancements in deep learning have demonstrated the efficacy of Convolutional Neural Networks (CNNs) in extracting and accurately classifying road line markings from vision data [4].

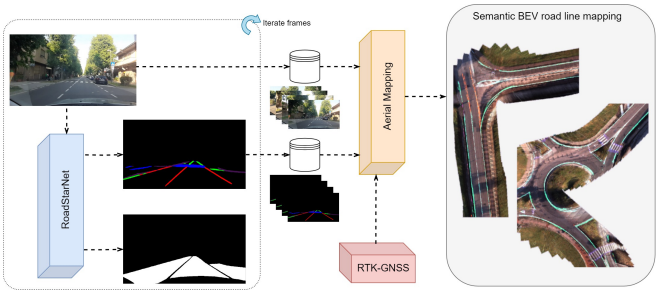


Fig. 1. High-level and simplified overview of our architecture. We propose a pipeline for frame-by-frame road line segmentation and subsequent precise GNSS data fusion to create aerial road line maps. Our multi-class segmentation network, RoadStarNet, identifies the road line markings and the drivable area; then, we obtain precise aerial segmented maps by aggregating and fusing data with RTK-GNSS locations.

The detection of lateral lines from sensory data is required in multiple automotive scenarios, with different degrees of required accuracy and computational performance. In ADAS, road line detection is important to determine the vehicle's position on the road and aid drivers in maintaining a safe driving experience. In these scenarios, cameras and simple CNNs can address the task, as commonly we are interested only in determining the position of the two closest lines just a few meters ahead. This task, moreover, can be performed on a frame-by-frame basis. In self-driving vehicles, instead, road line detection is a fundamental component for autonomous navigation, as it provides the vehicle with information on the road layout and allows it to make informed decisions. Here, a more complex representation is generally required. Accurate classification of the line types and precise detection up to a certain distance is typically a key requirement [5]. In these two scenarios, the architecture's design must ensure real-time performance. In HD map generation, instead, the detection of road lines is used to create precise digital maps of the road environment, providing crucial information for autonomous vehicles to make real-time decisions and safely navigate the roads. In this case, real-time requirements can be softened, as the map can be generated offline. At the same time, retrieving the most accurate representation of the road is fundamental, both in terms of segmented areas and correctly classified lines.

In this work, we propose a novel CNN-based architecture for the segmentation of road line markings in vehicle-acquired images. Furthermore, exploiting also Global Navigation Satellite System (GNSS) data, we show how this

This paper was supported by "Sustainable Mobility Center (Centro Nazionale per la Mobilità Sostenibile – CNMS)" project funded by the European Union NextGenerationEU program within the PNRR, Mission 4 Component 2 Investment 1.4. Also, this work was supported by AnteMotion S.r.l., Trento, Italy. Any opinions, findings, conclusions, or recommendations, either expressed or implied, in this material are those of the authors and do not necessarily reflect the views of the sponsoring organizations.

All authors are with the Department of Electronics, Information, and Bioengineering (DEIB), Politecnico di Milano, Milano, Italy, name.surname@polimi.it*, riccardoerminio.cortelazzo@mail.polimi.it†.

network is also suitable for generating accurate aerial images of road line markings, using our method simplified in Fig. 1. Indeed, the presented detector is the first stage of a pipeline for automatic semantic aerial map generation. To accurately train the proposed CNN, we improved the annotations of a state-of-the-art dataset [6] through label pre-processing allowing also for a systematic comparison of different road line segmentation algorithms. The proposed CNN uses a multi-decoder structure to perform multi-class road line segmentation and drivable area identification; the retrieved masks are then used by our mapping system, which projects the points extracted from the CNN into a Bird's-Eye View (BEV) space (i.e., a top-down view of the scene), and exploits the measurements of an accurate RTK-GNSS to georeference and integrate the extracted data. We validated our image segmentation pipeline on state-of-the-art annotated imagery data, and we experimentally evaluated our aerial mapping procedure on two real-world datasets. One, for a quantitative comparison, was acquired on the Monza race track, where accurate, manually-annotated ground truth is available. The other, instead, was acquired with a survey vehicle in an urban scenario and is more prone to a qualitative analysis. Experimental evaluations on real-world data, including data acquired with instrumented vehicles, reveal the effectiveness of the proposed pipeline in both frame-by-frame detection and integrated mapping quality.

To succinctly summarize the paper contributions, we now briefly outline the threefold core contribution of this paper. Firstly, we propose a novel CNN architecture for multi-class road line and drivable area segmentation, achieving state-of-the-art performance on the considered datasets. Secondly, we devise a processing technique to improve the road line marking annotations of the Berkeley Deep Drive 100k dataset (BDD100k) [6]. Thirdly, we show that our CNN is not suitable only for frame-by-frame prediction, but can also be effectively adopted for the generation of aerial road line maps through RTK-GNSS-camera fusion.

This paper is organized as follows. Related work is explored in Section II. The proposed CNN-based architecture for road line markings detection and classification, as well as drivable area identification, is presented in Section III, while in Section IV we illustrate our mapping technique to obtain accurate aerial views of the survey area and its road line markings. Finally, in Section V we experimentally evaluate our pipeline on real-world data, including data acquired with a vehicle equipped with a camera and an RTK-GNSS to validate the mapping pipeline.

II. RELATED WORK

In this section, we present a brief overview of related work on the segmentation and mapping of road line markings. The line detection pipeline is traditionally structured in multiple steps, as shown in [7]. The first component on which this work focuses is line detection on the image plane. This task has been performed using computer vision algorithms in the pre-deep-learning era. Gradient-based approaches leverage the sharp change in color between the pavement and the lateral line to perform detection. In particular, the Sobel operator and Canny algorithm have been popular choices for this task [8], [9]. But, these geometric approaches, while effective, were particularly subject to changes in illumination and ambient features. Therefore required strong assumptions to work with high accuracy. For this reason, deep-learning-based approaches have gradually gained popularity, also thanks to the constant release of big datasets to train the models, like CuLane [10] and BDD100k [6], consisting of a large number of annotated images. In particular, it has been possible to witness the usage of CNNs as a fallback mechanism to traditional gradient-based algorithms [11]. Chen et al. [12] presented a complete encoder-decoder network designed purely to perform lateral line segmentation from images. Recent works, like HybridNets [13], propose more complex architectures with multiple parallel decoders to perform road lines and drivable area detection. A similar approach is employed by RMNet [14] to perform multi-class road line detection. In particular, the CNN does not return a binary mask but also the type of lines and road paintings in the image. Recently, Garnett et al. [15] proposed an advancement to the classical line detection task, retrieving from a monocular camera also the 3D position of the road lines.

Moreover, joint approaches have been proposed, employing not only images acquired with a ground vehicle but also aerial images. Máttyus et al. [16] presented an innovative system on an extended version of the KITTI dataset [17], which combines images from the original dataset and aerial ones to perform accurate road segmentation. Aerial images can be used to generate lane-level HD maps [18], [19]. Similarly, Homayounfar et al. [20] perform road line detection on complex highway scenarios on a BEV space generated by combining high-resolution LiDAR scans to generate a detailed top view of the area of interest.

III. ROAD LINE MARKINGS RECOGNITION

In this section, we illustrate our CNN-based architecture used to address the task of detection and classification of road line markings, and the different loss functions considered during the training phase.

A. RoadStarNet Architecture

Recent work, with HybridNets [13], proposed a multi-decoder CNN architecture to address single-class line detection, drivable area identification, and object detection tasks. A dedicated decoder (segmentation head) deals with the first two segmentation-oriented tasks, while a second one (detection head) focuses on object detection only. Instead, the HybridNets shared encoder structure is composed of an EfficientNet-B3 [21] backbone network connected to a weighted Bi-directional Feature Pyramid Network (BiFPN) [22] module to extract multi-scale fused features from the input image. Indeed, BiFPN integrates features from various image resolutions, while computing weights, during the training phase, to estimate the relative importance

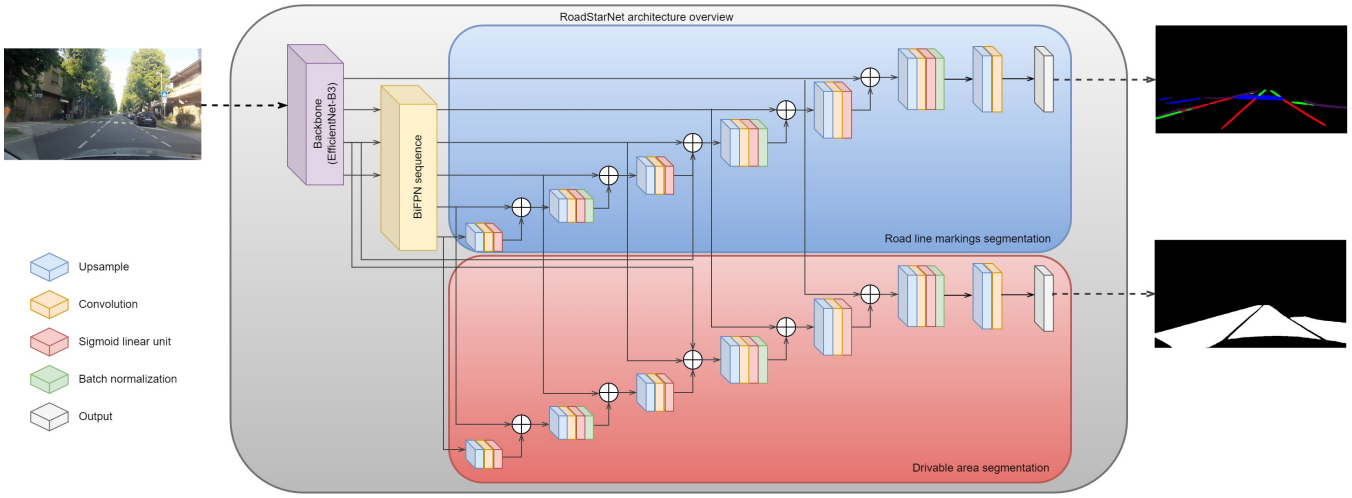


Fig. 2. Architecture overview of RoadStarNet. Given a camera image, our two-decoders CNN retrieves a multi-class segmentation of the road line markings, together with the drivable area.

of the levels. Their segmentation decoder resizes multi-scale features from BiFPNs via a series of upsamples and convolutions, combines them, and feeds them into an output module for segmentation prediction.

Inspired by HybridNets, we present our new CNN-based architecture to address multi-class line segmentation and drivable area identification. Similarly to RMNet [14] architecture, each of these tasks has its own dedicated decoder, with a U-Net-like [23] structure, while keeping the same shared encoder structure. Following the multi-task learning paradigm, the idea is to leverage shared representations to capture common patterns across a set of interconnected tasks [24], thus improving overall architecture performance. To retain additional spatial information that could potentially be lost during the BiFPN feature fusion process and enhance back-propagation, our RoadStarNet structure includes two backbone-decoders skip connections.

In RoadStarNet, feature blocks have an upsample layer followed by a convolutional layer with Sigmoid Linear Unit (SiLU) [25] as activation function in order to improve stability and avoid learning-related issues such as gradient exploding or vanishing. Moreover, to further improve model stability, the feature block series follows a pattern in which batch normalization is applied in alternating blocks. Intuitively, at each upscale step, the lower-level features are combined with the upper-level features by concatenating them. The network scheme thus composed is designed to exploit multi-scale features with various feature blocks handling different input dimensions. Finally, we added a post-processing phase in which clusters with a small size are discarded in an attempt to reduce the network output noise. Fig. 2 resumes and illustrates the proposed network architecture.

B. Training Loss Functions

Specific segmentation-oriented losses are used to train the CNNs presented in the literature. Define $\mathbf{T}_c$ as the Tversky index [26] for class $c \in \mathbf{C}$:

$$\mathbf{T}_c = \frac{\sum_{i=1}^{D} \mathbf{P}_{ci}\mathbf{G}_{ci}}{\sum_{i=1}^{D} \mathbf{P}_{ci}\mathbf{G}_{ci} + \alpha \sum_{i=1}^{D} \mathbf{P}_{ci}\mathbf{G}_{\bar{c}i} + \beta \sum_{i=1}^{D} \mathbf{P}_{\bar{c}i}\mathbf{G}_{ci}}, \tag{1}$$

where $\mathbf{C}$ is the set of classes, $\bar{c}$ denotes not belonging to class c, α and β are pre-defined constants to calibrate the magnitude of penalties for FPs and FNs, $\mathbf{P}$ and $\mathbf{G}$ are the prediction and ground truth (image) pixel-level data. The Tversky loss [27] is defined as:

$$L_T = \sum_{c \in \mathbf{C}} (1 - \mathbf{T}_c). \tag{2}$$

A second loss is also considered, the Focal loss [28]:

$$L_F = -\lambda \frac{1}{D} \sum_{c \in \mathbf{C}} \sum_{i=1}^{D} \mathbf{G}_{ci} \mathbf{P}_{\bar{c}i}^{\gamma} \log \mathbf{P}_{ci}, \tag{3}$$

where λ and γ are pre-defined constants. While the Tversky loss is more suitable when FPs and FNs need to be balanced, the Focal loss is more capable to learn hard labels as it down-weights the loss penalty of correctly classified targets.

In our work, we considered two losses to train our network. In particular, we considered a first loss defined, as in [13], as the sum of the Focal loss and the Tversky loss:

$$L_{F+T} = \xi_1 L_F + \xi_2 L_T, \tag{4}$$

where ξ_1 and ξ_2 are pre-defined constants. Also, to be able to weigh the various classes differently, we also considered a generalization of the Focal loss, which we define as Focal*:

$$L_{F*} = -\lambda \frac{1}{D} \sum_{c \in \mathbf{C}} \sum_{i=1}^{D} w_c \mathbf{G}_{ci} \mathbf{P}_{\bar{c}i}^{\gamma} \log \mathbf{P}_{ci}, \tag{5}$$

where w_c indicates an arbitrary pre-defined weight associated with class $c \in \mathbf{C}$. This loss allows us to weigh each class individually, making it possible to fine-tune the behavior of the network for a specific scenario. Within this paper, RoadStarNet-F* refers to the model trained using loss L_{F*}, and RoadStarNet-FT to the one trained using loss L_{F+T}.

C. Training Dataset

We train our model on the popular BDD100k [6] dataset, also used by state-of-the-art road line segmentation models [13], [29], [30]. As noted in the literature [29], however, in this dataset the lines are annotated in a non-straightforward way, i.e., polylines defining the two boundaries of each road line are available, and no line-boundaries match information is accessible. However, the dataset differentiates road lines by their line category (white or yellow, solid or dashed, zebra crossings, curbs, etc.). Such information in the form of polylines is hardly usable as-is for the task of image segmentation, where commonly image masks are used as targets. Rendering the annotated polylines as image masks, however, would not be suitable either. In such a way, the network would learn to identify the edges of each line, and not the center of the lines as desired. Post-processing the output of the network to extract the center of each line would also present difficulties, as such predictions are typically noisy.

The authors of YOLOP [29] proposed to address this issue with a processed version of the BDD100k annotations, which was then used by subsequent works, such as [13]. To this end, they provided binary image masks indicating the center of each road line instead of its edges. This allowed them to train their model as a standard segmentation network. The line width of such masks, however, is fixed-sized and cannot be customly changed to match different pre-defined widths (e.g., training set width). Moreover, the processed dataset lacks multi-class matching information, thus networks trained using this dataset cannot handle multi-class predictions. This ability is crucial, instead, when it is necessary to associate semantics to each line (e.g., different driving behavior in the presence of solid or dashed lines, or when facing a zebra crossing).

Although the introduction of these masks determined an important advancement in the field, we propose a further advancement with our data processing algorithm to generate multi-class segmentation masks, maintaining the class annotations present in BDD100k and allowing custom line width size in order to facilitate network comparisons. Note that parameters are customizable for future adaptation.

The algorithm works as follows. Given two line edges' points $\mathbf{L} = \{\mathbf{l}_1, ..., \mathbf{l}_n\}$ and $\mathbf{R} = \{\mathbf{r}_1, ..., \mathbf{r}_m\}$ of the same class in camera image coordinates, we first establish whenever the two sets belong to the same line. This is achieved by checking the proximity of $\mathbf{l}_1$ and $\mathbf{r}_1$, and likewise of $\mathbf{l}_n$ and $\mathbf{r}_m$. In other words, we check if:

$$||\mathbf{l}_1 - \mathbf{r}_1||_2 < \lambda(\mathbf{l}_1, \mathbf{r}_1)\delta \wedge ||\mathbf{l}_n - \mathbf{r}_m||_2 < \lambda(\mathbf{l}_n, \mathbf{r}_m)\delta, \quad (6)$$

where δ is a pre-defined constant and $\lambda : \mathbb{R}^2 \rightarrow [0, 1]$ is a scaling function. This scaling function is used to balance the maximum desired distance, i.e., when the two points are close to the horizon, the required maximum distance is shorter due to the perspective effects of the camera. Once a match is established, for each point $\mathbf{l}_i$ in $\mathbf{L}$, the closest point $\mathbf{r}_j$ in $\mathbf{R}$ is found and the average of their positions, $\mathbf{m}_i$, is taken:

$$\mathbf{m}_i = \frac{\mathbf{l}_i + \mathbf{r}_j}{2}. \quad (7)$$

The sets of $\mathbf{L}$ and $\mathbf{R}$ are finally replaced with $\mathbf{M} = \{\mathbf{m}_1, ..., \mathbf{m}_n\}$. This procedure generates a single edge for each line, which can be rendered into an image mask with lines of any given width. For coherence with the literature [29], we set this width to 8 px by default, although this parameter is customizable for future adaptations.

IV. SEGMENTED AERIAL MAPPING

In this section, we present our road line markings mapping pipeline to dynamically obtain top-down aerial views of the surveyed area.

A. IPM Model and BEVs Derivation

Generally, a sensor system acquires a variety of data that can be processed in a combined manner. The camera provides a sequence of N images $\{\mathbf{I}_i\}_{i=1,...,N}$, each acquired at time t_i; for each frame, the RTK-GNSS records the position of the vehicle and, thanks also to an inertial system and the fusion of the sensor data, it is possible to estimate the absolute location $\mathbf{L}_i$ and orientation $\mathbf{O}_i$ of the camera at the time of image acquisition. Pixels can be mapped into a relative 2D coordinate system thanks to Inverse Perspective Mapping (IPM) [31], which is used to get a BEV, namely an aerial top-down view of the scene. The standard IPM model consists of a 3×3 homography projection matrix $\mathbf{H}_i$ with 8 degrees of freedom (please refer to [32] for details), which describes the relationship between the camera view and the top-down view. Intrinsic and extrinsic camera parameters are required to determine $\mathbf{H}_i$: the former are generally constant and known as they depend on the type of camera, while the latter are dynamic due to the vehicle motion but can be estimated by integrating and fusing other sensors' data or by using calibration algorithms [33], [34]. Given a frame i, by combining the IPM model with $\mathbf{L}_i$ and $\mathbf{O}_i$, the road line pixels can be mapped in the world reference system.

B. Dynamic Block-Based Map Generation

Our map generation method is based on modular and extendable blocks. The proposed method allows a dynamic generation of map blocks saving memory and RAM consumption. It initially computes the total of map blocks, called chunks, to be generated. Intuitively, at each chunk corresponds a set of related camera images $\{\mathbf{I}_j\}_{j=h,...,k}$ whose vertices fall into the chunk space domain. A chunk can be thus derived by selecting a set of consecutive frames. The vertices of each image $\mathbf{I}_j$ (with $j \in \{h, ..., k\}$) are temporarily projected into the world coordinate system by leveraging RTK-GNSS data and the IPM model. This procedure allows the algorithm to compute both the dimension and the position of the chunks. Iteratively, given for each frame $\mathbf{H}_j$, $\mathbf{L}_j$, and $\mathbf{O}_j$, all pixels can be projected into the chunk map accordingly. At this point, road line pixels are substituted with their estimated road line marker class. Chunks can be merged together to derive the full map of the surveyed area.

TABLE I

EXPERIMENTAL RESULTS ON THE BDD100K DATASET BY CONSIDERING SINGLE CLASS AND 8 PX LINE DATA.

	Line Acc.	Line IoU	Dr.A. Acc.	Dr.A. IoU
RoadStarNet-F*	**82.44**	44.41	89.09	88.34
RoadStarNet-FT	66.50	53.45	87.89	87.66
YOLOPv2 [30]	80.11	53.24	91.31	**88.41**
HybridNets [13]	59.81	**53.82**	86.87	86.59
YOLOP [29]	65.40	49.70	**97.40**	86.00

V. EXPERIMENTS

All the experiments are conducted on a computer equipped with an Intel® Core™ i7-3770K processor and a NVIDIA® GeForce® GTX 970 GPU. All algorithms have been implemented in Python. To test our pipeline, we considered data composed of a large number of annotated images, as well as data acquired with a vehicle equipped with a standard survey sensors suite in different real-world environments.

A. Road Line Markings Recognition

We validate our model and compare it with state-of-the-art models from the literature using the test set of the BDD100k dataset [6]. The ground truth annotations we considered were generated through our processing pipeline (Section III-C). We evaluated the performance of our network on the detection of lines and drivable area, and we compared in Table I our results with state-of-the-art CNNs, including HybridNets [13], YOLOP [29], and YOLOPv2 [30]. From Table I, it is possible to notice how the proposed model achieves the highest accuracy on the line detection task while being almost on par with the compared approaches in Intersection over Union (IoU). The slight drop in this metric is justified by the fact that our network predicts slightly larger road lines. This characteristic does not significantly affect the final quality of the lines, but worsens the metric when the ground truth is instead narrow. Although the detection of the drivable area is not the focus of our work, and is only performed to aid our line detector, our system achieves acceptable results in terms of accuracy and IoU. Indeed, we report values close to compared models, with the exception of YOLOP, which however performed worse on line detection task. Lastly, Table II shows the classification performance obtained by our proposed method in terms of IoU; in the considered test, although the discrepancy is moderate, RoadStarNet-FT obtained better results with respect to RoadStarNet-F*.

We further show, in Fig. 3, a qualitative result obtained by testing our proposed methods on selected sample images from the CuLane dataset [10]. RoadStarNet-FT tends to be more conservative and therefore less prone to produce false positives, while RoadStarNet-F* identifies lines more decisively, thus being in our opinion more suitable for mapping purposes thanks to his ability to achieve a higher line mapping coverage.

(a) Ground truth (b) **RoadStarNet-F*** (c) **RoadStarNet-FT**

Fig. 3. Example of qualitative results: comparison of RoadStarNet single-class line detection output using different training functions on images from the CuLane dataset [10].

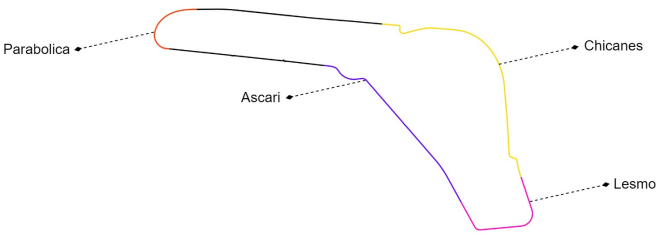


Fig. 4. Different areas of the Monza Eni Circuit track analyzed in our experimental validation (Table III): Chicanes (yellow), Lesmo (magenta), Ascari (purple), and Parabolica (orange).

B. Road Line Markings Mapping

The effectiveness of the mapping pipeline has been evaluated using two different datasets acquired in real-world scenarios. The first dataset was recorded at the Monza Eni Circuit (please refer to [35], [36] for detail), utilizing an autonomous vehicle equipped with imaging and positioning technologies, including cameras and an RTK-GNSS. The dataset comprises images and georeferenced ground truth positions of the lateral lines, which can be utilized to determine the system accuracy by comparing the output of the mapping pipeline (i.e., the lines projected in the BEV space) with manually mapped ones. To facilitate the comparison with other state-of-the-art techniques, the track was partitioned into multiple regions of interest (ROIs), depicted in Fig. 4, that include relevant hurdles to overcome (e.g., chicanes, tight corners).

Additionally, the system was tested in a more operational setting using a second dataset obtained in an urban environment with a multi-camera setup and an RTK-GNSS sensor. The acquisition took place in Tavagnacco, Italy, in

TABLE II

EXPERIMENTAL RESULTS (IOU) ON THE BDD100K DATASET BY CONSIDERING MULTIPLE CLASSES AND 8 PX LINE DATA.

RoadStarNet	single white solid	single white dashed	single yellow solid	single yellow dashed	double white solid	double white dashed	double yellow dashed	double yellow solid	crosswalk	road curb
RoadStarNet-FT	**50.50**	**50.26**	**54.58**	24.34	**25.90**	**12.04**	**60.79**	**33.99**	**48.40**	**42.06**
RoadStarNet-F*	43.75	42.99	48.32	**26.70**	23.16	11.81	51.11	30.70	41.31	37.79

the Adegliacco-Cavalicco area[1]. Unlike the circuit, manual mapping annotations of the line positions were not available in this scenario. Hence, the tests on this dataset serve as a qualitative demonstration of the pipeline's capability to accurately map urban areas, including challenging sections such as roundabouts and intersections.

We conducted our quantitative evaluation on the Monza Eni Circuit dataset. To evaluate the performance of our pipeline, two metrics were calculated. The first one, referred to as the mean prediction distance (Dist.), is computed as the mean distance between projected pixels and the center of the reference ground truth line[2]. This metric does not account, however, for the percentage of correctly predicted points. For this reason, a second metric to be considered simultaneously is introduced to gauge the coverage (Cov.), i.e., the percentage of the total road line markings actually mapped.

The results, presented in Table III, show how our RoadStarNet-F* obtained the highest coverage, albeit at the expense of a marginally higher mean prediction distance; this highlights the trade-off between the considered metrics. However, when considering the compared state-of-the-art model with the highest coverage, YOLOPv2, we notice that our network generally outperformed it in prediction distance, while also slightly enhancing its coverage. When referring to the considered tests, our model decisively identified road line pixels. This property is clearly highly desirable for aerial mapping, as it allows to extract more road markings information from a single vehicle survey. Our alternative model, RoadStarNet-FT, instead, achieved prediction distances among the lowest, while providing a satisfactory mapping coverage level. For this reason, its trade-off represents a viable alternative to RoadStarNet-F* when a slight relaxation of mapping criteria is deemed acceptable. In comparison, HybridNets, while achieving the lowest prediction distance, displays a highly conservative behavior and does not provide a satisfying coverage, mapping less than 65-75% of several sections. This behavior can be observed also in Fig. 5, showing a visual comparison of the predictions of each network.

Qualitative result examples are shown in Fig. 6, presenting the reconstructed aerial and semantic maps of selected challenging areas in the considered urban environment of Tavagnacco, including roundabouts and intersections. Results suggest the effectiveness of our proposed pipeline, specifically with RoadStarNet-F* as CNN, and its ability to detect and map road line markings in the area. Different road line markings' colors indicate different classes; Fig. 6 shows also that the CNN classified the points with considerable accuracy, including crosswalks.

[1]Coordinates: 46°06'43.3"N 13°13'49.4"E.

[2]Notice that, as the road line markings have a non-zero width, not represented in the ground truth, this metric overestimates the error.

VI. CONCLUSIONS

In this paper, we introduced a new architecture for the recognition and mapping of road line markings to obtain accurate aerial images of the whole survey area. The proposed CNN uses a multi-decoder to perform multi-class segmentation of images acquired by a vehicle-mounted camera. The segmentation masks and the images are then projected into a BEV space and combined with RTK-GNSS data to reconstruct an aerial view of the traveled area with information about road line markings. We also devised a pre-processing algorithm for the popular BDD100k dataset to generate a more suitable ground truth for road lines. The proposed model has been tested both on the public dataset BDD100k against state-of-the-art models and on datasets acquired in real-world environments to evaluate the whole mapping pipeline. Experimental results show the effectiveness of the proposed CNN-based pipeline and how is well-suited for the reconstruction of complete segmented aerial maps of large areas.

REFERENCES

[1] A. Moujahid, M. E. Tantaoui, M. D. Hina, A. Soukane, A. Ortalda, A. ElKhadimi, and A. Ramdane-Cherif, "Machine learning techniques in ADAS: A review," in *Proc. of International Conference on Advances in Computing and Communication Engineering*, 2018, pp. 235–242.

[2] K. Muhammad, A. Ullah, J. Lloret, J. Del Ser, and V. H. C. de Albuquerque, "Deep learning for safe autonomous driving: Current challenges and future directions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4316–4336, 2021.

[3] S. P. Narote, P. N. Bhujbal, A. S. Narote, and D. M. Dhane, "A review of recent advances in lane detection and departure warning system," *Pattern Recognition*, vol. 73, pp. 216–234, 2018.

[4] A. A. Mamun, E. P. Ping, J. Hossen, A. Tahabilder, and B. Jahan, "A comprehensive review on lane marking detection using deep neural networks," *Sensors*, vol. 22, no. 19, p. 7682, 2022.

[5] S. Arrigoni, S. Mentasti, F. Cheli, M. Matteucci, and F. Braghin, "Design of a prototypical platform for autonomous and connected vehicles," in *Proc. of AEIT International Conference on Electrical and Electronic Technologies for Automotive*, 2021, pp. 1–6.

[6] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving video database with scalable annotation tooling," *arXiv preprint arXiv:1805.04687*, 2018.

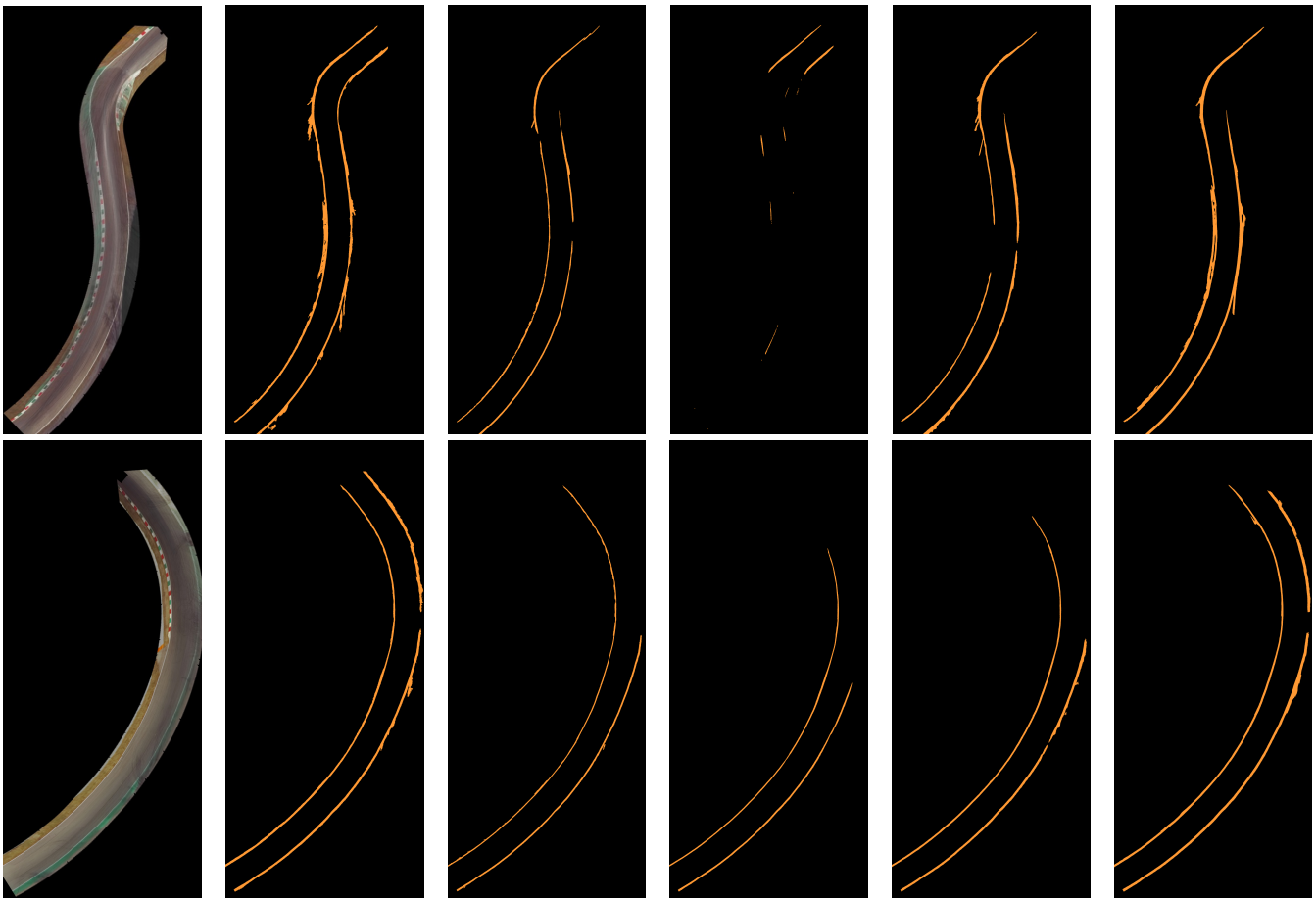
[7] P. Cudrano, B. Gallazzi, M. Frosi, S. Mentasti, and M. Matteucci, "Clothoid-based lane-level high-definition maps: Unifying sensing and control models," *IEEE Vehicular Technology Magazine*, vol. 17, no. 4, pp. 47–56, 2022.

[8] K. H. Lim, K. P. Seng, L.-M. Ang, and S. W. Chin, "Lane detection and Kalman-based linear-parabolic lane tracking," in *Proc. of International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2009, pp. 351–354.

TABLE III

MEASURED MEAN PREDICTION DISTANCE AND COVERAGE ON THE CONSIDERED SECTIONS OF THE MONZA ENI CIRCUIT TRACK: NORMAL / LIGHT POST-PROCESSING OUTPUT CLEANING. COVERAGE PARAMETER IS ROUNDED DOWN.

	Sections								Circuit	
	1: Chicanes		2: Lesmo		3: Ascari		4: Parabolica			
Model	Dist. (m)	Cov. (%)	Dist. (m)	Cov. (%)	Dist. (m)	Cov. (%)	Dist. (m)	Cov. (%)	Dist. (m)	Cov. (%)
RoadStarNet-F*	0.40 / 0.43	**98 / 98**	0.63 / 0.71	**99 / 99**	0.55 / 0.58	**99 / 99**	0.48 / 0.50	**98 / 98**	0.56 / 0.60	**99 / 99**
RoadStarNet-FT	**0.29 / 0.30**	83 / 90	0.42 / 0.40	83 / 94	0.38 / 0.38	90 / 96	0.38 / 0.39	80 / 93	0.42 / 0.43	88 / 94
YOLOPv2 [30]	0.37 / 0.46	94 / 96	0.73 / 0.79	98 / 99	0.60 / 0.64	98 / 99	0.45 / 0.46	97 / 98	0.57 / 0.66	97 / 98
HybridNets [13]	0.32 / 0.34	85 / 93	**0.32 / 0.33**	61 / 69	**0.34 / 0.34**	64 / 74	**0.31 / 0.34**	64 / 74	**0.35 / 0.38**	76 / 83
YOLOP [29]	0.29 / 0.38	86 / 93	0.42 / 0.44	90 / 95	0.42 / 0.43	93 / 97	0.41 / 0.45	78 / 92	0.42 / 0.47	90 / 95



(a) Reference BEV (b) **RoadStarNet-F*** (c) **RoadStarNet-FT** (d) HybridNets [13] (e) YOLOP [29] (f) YOLOPv2 [30]

Fig. 5. Comparison of the mapping pipeline using different segmentation models on two portions of the Monza Eni Circuit track. Our proposed architectures (in bold) display a high coverage while maintaining acceptable distance error.

[9] Y. Wang, D. Shen, and E. K. Teoh, "Lane detection using spline model," *Pattern Recognition Letters*, vol. 21, no. 8, pp. 677–689, 2000.

[10] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatial CNN for traffic scene understanding," in *Proc. of AAAI Conference on Artificial Intelligence*, 2018, pp. 7276–7283.

[11] J. Kim and M. Lee, "Robust lane detection based on convolutional neural network and random sample consensus," in *Proc. of International Conference on Neural Information Processing*, 2014, pp. 454–461.

[12] P.-R. Chen, S.-Y. Lo, H.-M. Hang, S.-W. Chan, and J.-J. Lin, "Efficient road lane marking detection with deep learning," in *Proc. of IEEE International Conference on Digital Signal Processing*, 2018, pp. 1–5.

[13] D. Vu, B. Ngo, and H. Phan, "HybridNets: End-to-end perception network," *arXiv preprint arXiv:2203.09035*, 2022.

[14] W. Jang, J. Hyun, J. An, M. Cho, and E. Kim, "A lane-level road marking map using a monocular camera," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 1, pp. 187–204, 2021.

[15] N. Garnett, R. Cohen, T. Pe'er, R. Lahav, and D. Levi, "3D-LaneNet: End-to-end 3D multiple lane detection," in *Proc. of IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2921–2930.

[16] G. Máttyus, S. Wang, S. Fidler, and R. Urtasun, "HD maps: Fine-grained road segmentation by parsing ground and aerial images," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3611–3619.

[17] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[18] G. Máttyus, S. Wang, S. Fidler, and R. Urtasun, "Enhancing road maps by parsing aerial images around the world," in *Proc. of IEEE International Conference on Computer Vision*, 2015, pp. 1689–1697.

[19] Y. Wei, F. Mahnaz, O. Bulan, Y. Mengistu, S. Mahesh, and M. A. Losh, "Creating semantic HD maps from aerial imagery and aggregated vehicle telemetry for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 382–15 395, 2022.

[20] N. Homayounfar, W.-C. Ma, J. Liang, X. Wu, J. Fan, and R. Urtasun,

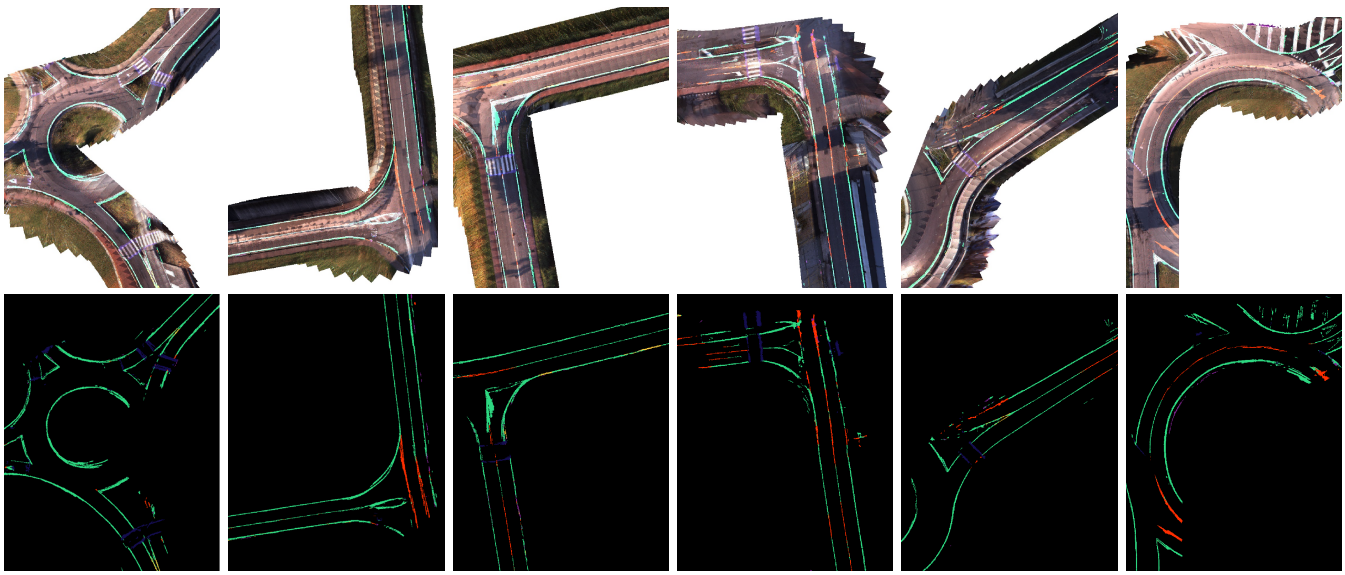


Fig. 6. Examples of qualitative results representing aerial views of road line markings obtained via the proposed pipeline. Green color refers to solid lines, while red color refers to dashed lines. Roundabouts are partially occluded because the path of the survey vehicle only covered part of it.

"DAGMapper: Learning to map by discovering lane topology," in *Proc. of IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2911–2920.

[21] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. of International Conference on Machine Learning*, 2019, pp. 6105–6114.

[22] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 781–10 790.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. of International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.

[24] M. Crawshaw, "Multi-task learning with deep neural networks: A survey," *arXiv preprint arXiv:2009.09796*, 2020.

[25] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, vol. 107, pp. 3–11, 2018.

[26] A. Tversky, "Features of similarity," *Psychological Review*, vol. 84, no. 4, p. 327, 1977.

[27] S. S. M. Salehi, D. Erdogmus, and A. Gholipour, "Tversky loss function for image segmentation using 3D fully convolutional deep networks," in *Proc. of International Workshop on Machine Learning in Medical Imaging*, 2017, pp. 379–387.

[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. of IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.

[29] D. Wu, M.-W. Liao, W.-T. Zhang, X.-G. Wang, X. Bai, W.-Q. Cheng, and W.-Y. Liu, "YOLOP: You only look once for panoptic driving perception," *Machine Intelligence Research*, vol. 19, no. 6, pp. 550–562, 2022.

[30] C. Han, Q. Zhao, S. Zhang, Y. Chen, Z. Zhang, and J. Yuan, "YOLOPv2: Better, faster, stronger for panoptic driving perception," *arXiv preprint arXiv:2208.11434*, 2022.

[31] H. A. Mallot, H. H. Bülthoff, J. J. Little, and S. Bohrer, "Inverse perspective mapping simplifies optical flow computation and obstacle detection," *Biological Cybernetics*, vol. 64, no. 3, pp. 177–185, 1991.

[32] Y. Chen, Z. Xiang, and W. Du, "Improving lane detection with adaptive homography prediction," *The Visual Computer*, pp. 1–15, 2022.

[33] J. Jeong and A. Kim, "Adaptive inverse perspective mapping for lane map generation with SLAM," in *Proc. of International Conference on Ubiquitous Robots and Ambient Intelligence*, 2016, pp. 38–41.

[34] M. Bellusci and M. Matteucci, "Advances in real-time online vehicle camera calibration via road line markings parallelism enforcement," in *Proc. of IEEE Intelligent Vehicles Symposium*, 2022, pp. 1511–1516.

[35] P. Cudrano, S. Mentasti, M. Matteucci, M. Bersani, S. Arrigoni, and F. Cheli, "Advances in centerline estimation for autonomous lateral control," in *Proc. of IEEE Intelligent Vehicles Symposium*, 2020, pp. 1415–1422.

[36] M. Bersani, S. Mentasti, P. Cudrano, M. Vignati, M. Matteucci, and F. Cheli, "Robust vehicle pose estimation from vision and INS fusion," in *Proc. of International Conference on Intelligent Transportation Systems*, 2020, pp. 1–6.