Model Guided Road Intersection Classification

Augusto Luis Ballardini[1], Álvaro Hernández Saz and Miguel Ángel Sotelo

***Abstract*— Understanding complex scenarios from in-vehicle cameras is essential for safely operating autonomous driving systems in densely populated areas. Among these, intersection areas are one of the most critical as they concentrate a considerable number of traffic accidents and fatalities. Detecting and understanding the scene configuration of these usually crowded areas is then of extreme importance for both autonomous vehicles and modern Advanced Driver Assistance Systems (ADAS), aimed at preventing road crashes and increasing the safety of Vulnerable Road Users (VRU). This work investigates how to classify intersection areas from RGB images using well-consolidated neural network approaches along with a method to enhance the results based on the teacher/student training paradigm. An extensive experimental activity aimed at identifying the best input configuration and evaluating different network parameters on both the well-known KITTI dataset and the new KITTI-360 sequences shows that our method outperforms current state-of-the-art intersection classification approaches on *per-frame* basis, proving the effectiveness of the proposed learning scheme.**

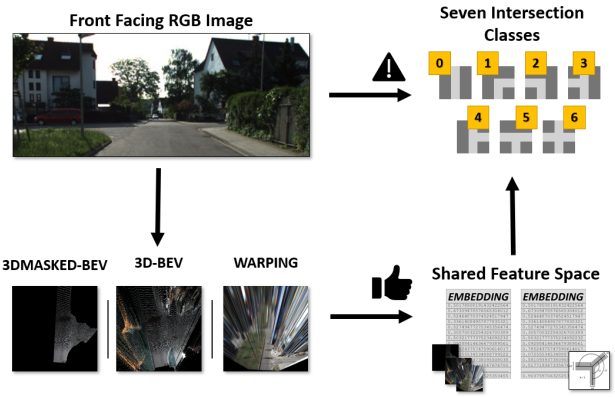


Fig. 1: A short overview of the proposed classification method. Our proposal exploits a synthetic intersection generator to enhance the prediction over standard RGB images, following a teacher/student training scheme.

I. Introduction

Estimating the scene in front of a vehicle is crucial for safe autonomous vehicle maneuvers, and it is also key to advanced ADAS. Even though performance and availability of scene understanding systems increased over the past years, technology seems to be far from the requirements of SAE full-automation level, particularly regarding urban areas and contexts without a strict Manhattan-style city planning.Among these, intersection areas are one of the most critical, and reports from the United States National Highway Traffic Safety Administration (NHTSA) show us that intersections concentrate more than 40% of motor vehicle crashes [1]. Navigation in these areas requires robust systems to correctly identify them, enabling safe maneuvers as the vehicle approaches and crosses the upcoming intersection. From an opposite viewpoint, it follows that we can exploit the detection and classification of intersections as input to high-level classifiers of driver's maneuvers, or to ease the prediction of position and intentions of VRU. Toward this goal, some intersection detectors are tightly coupled with localization procedures that in turn rely on external systems such as Global Navigation Satellite System (GNSS) or map providers like Google Maps, HERE, or TomTom, which started to provide maps commonly referred to as High Definition maps (HD maps). The benefits of having prior knowledge about the road configuration from maps are undisputed. It allows systems to narrow the localization uncertainty and the plethora of driving scenarios, hence exploiting the map data to perform predefined tactical and operational maneuvers. However, given the impact of the vehicle crashes, it follows that relying on updated maps might jeopardize the safety of autonomous driving systems themselves. Moreover, GNSS reliability in urban areas is frequently hampered by multi-path or non-line-of-sight (NLOS) issues, requiring self-sustaining approaches and on-board sensors. State-of-the-art intersection detection algorithms use a combination of techniques ranging from consolidated computer vision approaches to probabilistic methods to jointly process 3D data from Light Detection And Ranging (LiDAR) sensors, images and map features. Nevertheless, research progresses during the past years on Deep Neural Networks (DNNs) outperformed previous proposals on almost every task, ranging from stereo reconstruction to object detection and image segmentation, according to [2]–[5].

This project aims to exploit the generalization capabilities of modern DNNs pushing forward the edges in road intersection classification context. Our intent is to understand the typology of intersections in front of a vehicle from RGB sensors only, assessing whether a single-frame technique may serve this purpose and highlighting the limitations. This aims to support a wide range of advanced driving assistance systems (ADAS) and self-driving algorithms, which can significantly benefit from the intersection classification for

All authors are from Computer Engineering Department, Polytechnic School, Universidad de Alcalá, Alcalá de Henares, Spain.

[1]His work has been funded by European Union H2020, under GA Marie Skłodowska-Curie n. 754382. The content of this work does not reflect the official opinion of the European Union. Responsibility for the information and views expressed in therein lies entirely with the author. {augusto.ballardini, alvaro.hernandezsaz, miguel.sotelo}@uah.es

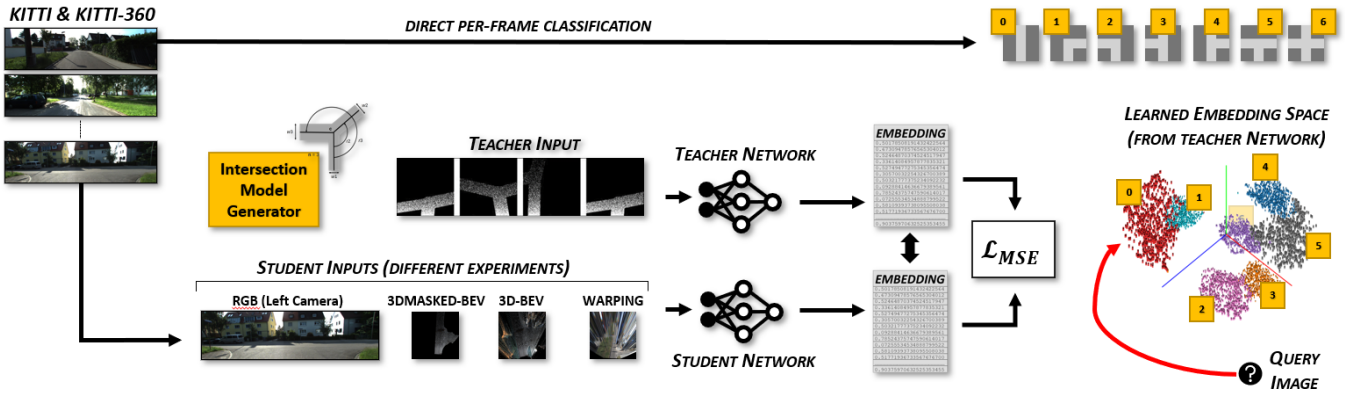


Fig. 2: A schematic description of the performed activities. The upper section depicts the basic classification pipeline by means of well-consolidated DNNs. In the lower section, the schematic of the teacher/student learning paradigm applied to the intersection classification problem, where two DNNs are trained to obtain similar embedding vectors. The rightmost question mark represents a query using the different inputs (see student inputs).

many sub-task, such as localization purposes. Following our previous works in intersection classification [6], [7] and road segmentation [8], we propose to identify the intersection classes shown in Figure 1. Despite the limitation on the seven classes, this allows us to compare the improvements concerning the previous state-of-the-art, yet paving the way for further investigation on other intersection configurations, *e.g.,* roundabouts or ample avenues. Moreover, differently from previous approaches, our proposal can predict the intersection on a *per-frame* basis without any temporal integration, exceeding the previous state-of-the-art resulting accuracy. Specifically, we demonstrate that our proposal effectively exploits both kitty [9] and new KITTI-360 sequences [10].

II. Related Work

The relevance of road intersection detection can be noticed from the interest towards the problem coming from different research communities as well as traffic regulation agencies [11], [12]. From a technical perspective, we can first distinguish approaches that exploit images from both stereo or monocular camera-suites, algorithms that only rely on LiDAR sensors, or finally, a combination of the previous. The first research that appeared in the intersection detection field dates back to the '80s and the works of Kushner and Puri [13], where matches between road boundaries extracted from images or LiDAR sensors data and the correspondent topological maps were exploited to detect intersection areas. In different contexts but with a similar approach, authors in [6], [14] exploited RGB images from vehicle front-facing cameras and standard computer vision techniques to create temporally integrated occupancy grids that were in turn compared to predetermined shapes to assess the presence of upcoming intersections. During the same period, the authors in [15] proposed a method where three different classifiers were evaluated to distinguish *junctions* from *roads*. Different from the previous method, here, 3D point clouds from LiDARs were used. A second distinction can be made for works involving deep-learning techniques. Works in this category include the approach in [16] with a network called *IntersectNet*, where a sequence of 16 images was passed through an ensemble of Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), combined to set up a three-type intersection classifier (four road-crossing and T-junctions) using a simple average-pooling fusion layer. A similar ensemble coupled with a more elaborated integration network was used in the work proposed in [17]. Here the authors suggest using two sets of images relative to the intersection processed with DNN and RNN, respectively. Regarding the LiDAR domain, the authors in [18] proposed a network called LMRoadNet aimed to simultaneously segment road surface and perform topology recognition by an aggregation of consecutive measures. Another interesting approach that exploits LiDAR was presented in [19]. Here the authors evaluated a transfer-learning method to the classification problem, coupling the sensor readings with the prediction of the ego-vehicle path. Unlike the previous approaches, we propose a method for classifying upcoming intersections using the *Teacher-Student* learning paradigm, where a combination of two networks is employed. The main contribution of this work is then to assess whether this learning method can be used to *guide* the learning of a specific task between different models, usually but not limited to simpler ones. In particular, we propose to exploit the simple yet previously assessed intersection model generator presented in [6] as input to the teacher network and then evaluate the learning capabilities of different student network configurations.

III. Technical Approach

Our intention consists in identifying the topology of the upcoming intersection. We used selected sequences from two datasets collected in two different periods in Karlsruhe, Germany [9], [10]. The temporal distance spans over two years; thus, we believe it is fair to say that the scenes appear different enough to stress our system generalization capabilities. Regarding the frame selection, we have to distin-

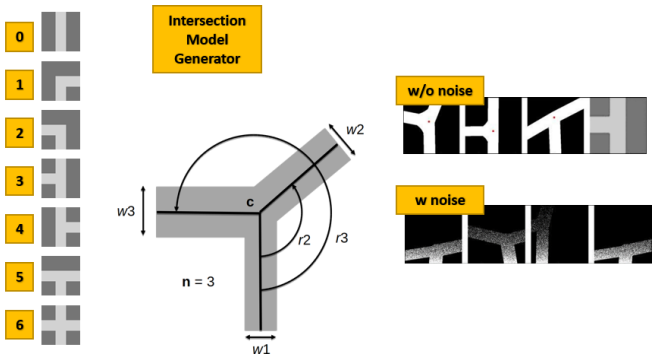


Fig. 3: The seven intersection classes along with the model used to generate the training dataset. In the two following lines: a triplet consisting of two samples of the canonical type-0 (shown in the last box of the row) and a different one, *e.g.,* type-5, and an example of the application of the random noise.

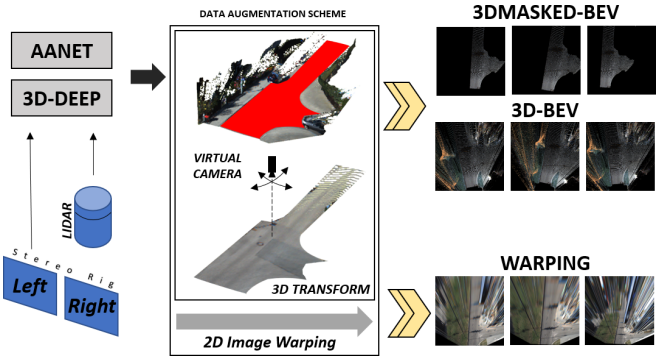


Fig. 4: The figure depicts the pipeline used to generate the images used in this work (apart from the original RGB-left camera). Among them, only the 3DMASKED-BEV use the LiDAR data.

guish the selection process between the two datasets. On the one hand, KITTI is distributed with GPS-RTK ground-truth data that allows us to exploit the localization information to automatically select and classify the frames involving intersections, exploiting the cartography of OpenStreetMap. Besides speeding up the process, this allowed us to use frames up to a specific distance from the intersection center. Specifically, we used the selected frames used in [20]. On the other hand, the new KITTI-360 dataset is currently missing the promised OpenStreetMap data and per-frame GPS-like positioning. This forced us to perform a frame labeling relying only on the appearance of each frame. Nevertheless, we were able to manually label all ten sequences of the dataset used, alternatively, to train and test the performances of our approach. Further details will be provided in Section III-D.3. The underlying idea of this work is twofold. First, we wanted to prove the capabilities of the teacher/student paradigm in identifying the upcoming intersections, with respect to a basic baseline composed of standard state-of-the-art neural networks. Toward this goal, different approaches were experimented and will be described in the following subsections. Second, this work assesses the classification capabilities of such networks on a frame-by-frame basis, to compare the multi-frame results of previous contributions with the proposed learning paradigm. An overview of the entire pipeline described in this work is proposed in Figure 2, and the following subsections explore the extensive experimental activity performed towards our goals.

A. RGB Pre-processing

To facilitate the comparison concerning the Model-Based Bird Eye Views (MBEVs) images generated with the intersection model of Section III-A, we created a pipeline that allows us to transform the RGB images into a similar viewpoint. Due to the low amount of frames selected from the first KITTI dataset, we opted for the following scheme, allowing for simultaneous data augmentation and bird-eye-view image transforms. First, using the work in [21] and both the images from the stereo-rig, we created the associated depth-image, which allows us effortlessly to generate a 3D point cloud of the observed scene. We then apply to the depth-image the road segmentation mask obtained using the algorithm presented in [8], to remove the 3D points that do not belong to the road surface. The remaining 3D points are then used to create the so-called Masked 3D-generated Bird Eye Views (3DMASKED-BEVs), which in turn are very similar to those generated using the intersection model. Please note that having the 3D coordinates allows us to easily generate as many views as needed even from a single stereo pair, fulfilling the common data-augmentation needs for neural network approaches, see Figure 4. This allowed us to emulate the sparsity issue of the 3DMASKED-BEVs. Eventually, to measure the contribution of this additional information to the classification problem, we also generated a version of the images without applying the segmentation mask. We refer to these images as 3D-generated Bird Eye Views (3D-BEVs). An example of all the possible outcomes resulting from one original single pair of images is shown in Figure 4.

B. Intersection Model

From a technical perspective and among the possible use cases, the idea behind the teacher/student training paradigm includes transferring knowledge from a more simple domain to a much more complex one. In our case, the base domain from which we propose to learn consists of a synthetic set of Bird Eye View (BEV) images generated with the intersection model used in the works [6], [7] for intersection classification and vehicle localization, respectively. The simple intersection model generator, along with the seven configuration classes, is visually described in Figure 3. Its complete parameterization includes the possibility to change not only the intersection typology, *e.g.,* the number and position of intersecting arms, but also the width of each individual road and the center position with respect to the image. This model allows us to generate BEV binary images containing the shape of all the considered intersections types that can be found in the two datasets and an arbitrary amount of them. During the training phases of our teacher network, these will be used, acting itself as a data-augmentation

scheme for the DNN. We refer to these images as MBEVs. At this time, despite its triviality, it should be noted that the point-density of 3DMASKED-BEVs is not constant over the distance with respect to the vehicle. Therefore, to simulate comparable MBEVs, we added a random noise proportional to the distance, see Figure 3.

C. Baseline

We started our experiments by evaluating the classification capabilities of two well-known network models, namely RESNET-18 and VGG-11 networks, to perform classification in an end-to-end fashion. This allowed us to create an architecture baseline to compare with. Please notice that these networks will then be used as the backbone for all subsequent activities. To create this baseline, we first used the RGB images from the left camera of the stereo rig. Alongside, we also prepared a second set of images containing a 2D-homographies of the original images to obtain so-called warpings with homography (WARPING) images. These two sets of images, together with the previous RGB images, were used to perform a comparison between the two representations, then assess the benefits described in Section I. It is worth mentioning that a fair comparison with most existing approaches at this stage is not possible, as they used a frame-integration process. Nevertheless, this helped us to set lower-bound thresholds and to evaluate the approaches described in the following subsections

D. Teacher/Student Training

In order to compare the images generated from the intersection model and those transformed from the RGB images, we propose a teacher/student paradigm aimed to learn a shared-embedding space between the two domains. The approach proposed in this work is inspired by the work of Cattaneo [22], which performs visual localization using 2D and 3D inputs in a bi-directional mode, teaching two networks to create a shared embedding space. In a similar way, we conceive the classification problem as a metric-learning task where, given two instances of the same intersections class but in different domains, *e.g., Class 0* and *Domains D1* and *D2* ($D^1_{C=0}$ and $D^2_{C=0}$), and two different non-linear functions $f(\cdot)$ and $g(\cdot)$ represented in the form of DNNs, the distance between the embeddings is lower than any other negative intersection instance, *e.g.,* $D^2_{c=2}$. Formally, given the *Intersection-Model* domain $M = \{0, 1, ..., 6\}$ and *Camera* domain $C = \{0, 1, ..., 6\}$ each of which contains the seven intersection typologies considered in Section III-A, given one element $m_i \in M$, then Equation (1) is satisfied for all the elements in $c_j \in (C \setminus c_i)$, where $d(\cdot)$ is a distance function.

$$d(f(m_i), g(c_i)) < d(f(m_i), g(c_j)) \tag{1}$$

With regard to the teacher/student learning scheme, we made the following considerations.

1) Teacher: The teacher network is the first of the two networks to be trained. It uses the images generated from the intersection model to create a high-dimensional embedding vector associated with each of the seven intersection typologies. We used a triplet margin approach [23], where a set of three images generated with the intersection model (M^a_i, M^s_i, M^d_i) composed of one *anchor* class image M^A_i, a *same* class sample M^S_i and a *different* class sample M^D_i, is passed through the triplet margin loss function. The function is defined similarly to each part of Equation (1), but this time using the same DNN model $f(\cdot)$, *i.e.,* our teacher network, as follows:

$$\begin{aligned}\mathcal{L} = \sum_i [&d(f(M^A_i), f(M^S_i)) \\ &-d(f(M^A_i), f(M^D_i)) + m]_+\end{aligned} \tag{2}$$

where $[\cdot]_+$ means $max(0, [\cdot])$ and $d(x_i, y_i) = \|x_i - y_i\|_p$ with p as the norm degree for pairwise distance, that in our case was set to L^2. As we desire the seven embedding vectors be as much separated as possible, a high separation margin value m was used. Figure 5 depicts the resulting separation.

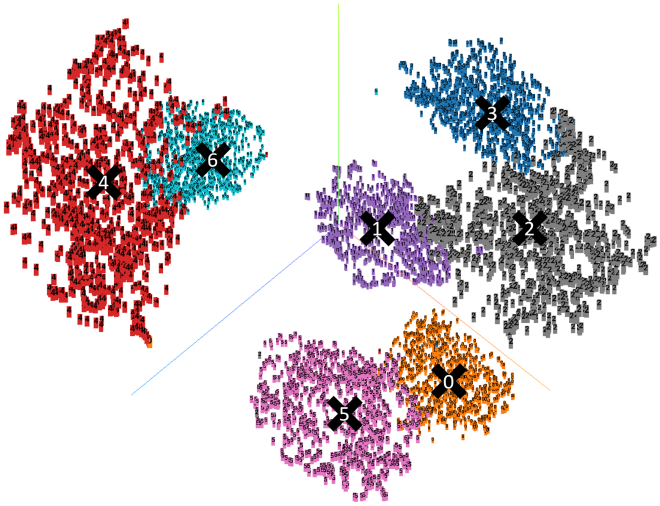


Fig. 5: The embedding space visually represented using T-SNE algorithm. In black, we conceptually represent the centroid of each of the clusters.

2) Student: Once the teacher has been trained, we trained the student network using the pre-processed RGB images as input data in a way to obtain a similar embedding vector. Towards this goal, the loss-function is composed as follows:

$$\mathcal{L} = \sum_i [d(f(M^A_i), f(C^S_i))] \tag{3}$$

where M and C are the model-domain and camera-domain as previously stated and MSE was used as distance function $d(\cdot)$ between the embeddings. It is worth mentioning that to maintain a consistent distance within same-class classifications, M^A_i elements were chosen not from the list of embedding vectors used in the training phase of the teacher network, but rather from the average of 1000 new random samples generated after the teacher network was trained, *i.e.,* never seen before from the DNNs. These per-class averages, *i.e.,* cluster centroids, are shown in Figure 5 with black crosses, and represent therefore our M^A_i set.

TABLE I: Overall Accuracy Results

Single Sequence Results			**KITTI**				**KITTI-360**		
			3DMASKED-BEV (*)	3D-BEV	WARPING	RGB	3D-BEV	WARPING	RGB
Baseline	Resnet18	C.E.	✗	✗	0.411	0.343	✗	0.633	0.426
		F.C.	✗	✗	0.467	0.345	✗	0.629	0.452
	VGG11	C.E.	✗	✗	0.409	0.352	✗	0.729	0.562
		F.C.	✗	✗	0.390	0.344	✗	0.673	0.566
Ours	Resnet18	MSE	0.723	0.334	**0.514**	**0.401**	0.677	**0.745**	0.563
	VGG11	MSE	0.687	0.221	0.381	0.315	0.602	**0.752**	0.456
Cross Dataset Results			**KITTI**				**KITTI-360**		
			3DMASKED-BEV (*)	3D-BEV	WARPING	RGB	3D-BEV	WARPING	RGB
Baseline	Resnet18	C.E.	✗	✗	0.410	0.316	✗	0.599	0.597
		F.C.	✗	✗	0.414	0.303	✗	0.558	0.576
	VGG11	C.E.	✗	✗	0.417	0.327	✗	0.621	0.609
		F.L.	✗	✗	0.383	0.335	✗	0.625	0.625
Ours	Resnet18	MSE	0.723	0.315	**0.449**	**0.346**	0.447	**0.640**	0.615
	VGG11	MSE	0.687	0.241	0.333	0.230	0.282	0.630	0.516

(*) Please note that these results were obtained using the validation set. F.C.: focal-loss. C.E.: Cross Entropy loss. The metrics in *baseline* rows differs from those in *ours* as we tackled the classification problem as a metric learning task (see further details in Section III-C).

3) Training Details: A data augmentation process was introduced in both networks to avoid overfitting during the network's training phase. We generated a set of 1000 per-class intersection configurations by sampling from our generative model for what concerns the teacher network. We applied a normal random noise to the seven *canonical* intersection configurations on each parameter involved in the generation of the intersection, *e.g.,* width, angle and intersection center, in a measure of [2.0m, 0.4rad, 9.0m]. For what concerns the noise, starting from the bottom of the image, we added an increasing number of random noise to each line in a way to mimic the 3D density effect of 3DMASKED-BEVs. Regarding the student network, since the low number of intersections present in the two KITTI datasets in comparison with the overall number of frames, we performed data augmentation adding a 6-DoF displacement to a looking-down virtual camera initially set at $[10m, 22.5m]$ above the road surface and $[17, 22m]$ in front of the vehicle for the KITTI and KITTI-360 respectively. Due to the nature of type-1 and type-2 intersection classes, which contains any kind of curve without a specific curvature threshold, we zeroed the rotation along the vertical axis to limit the chance of assimilating these samples to the type-0 class. Our code leverages the PyTorch 1.6 learning framework [24] and both teacher and student images were scaled to images with size 224x224 pixels.

IV. Experimental Results

A. Dataset

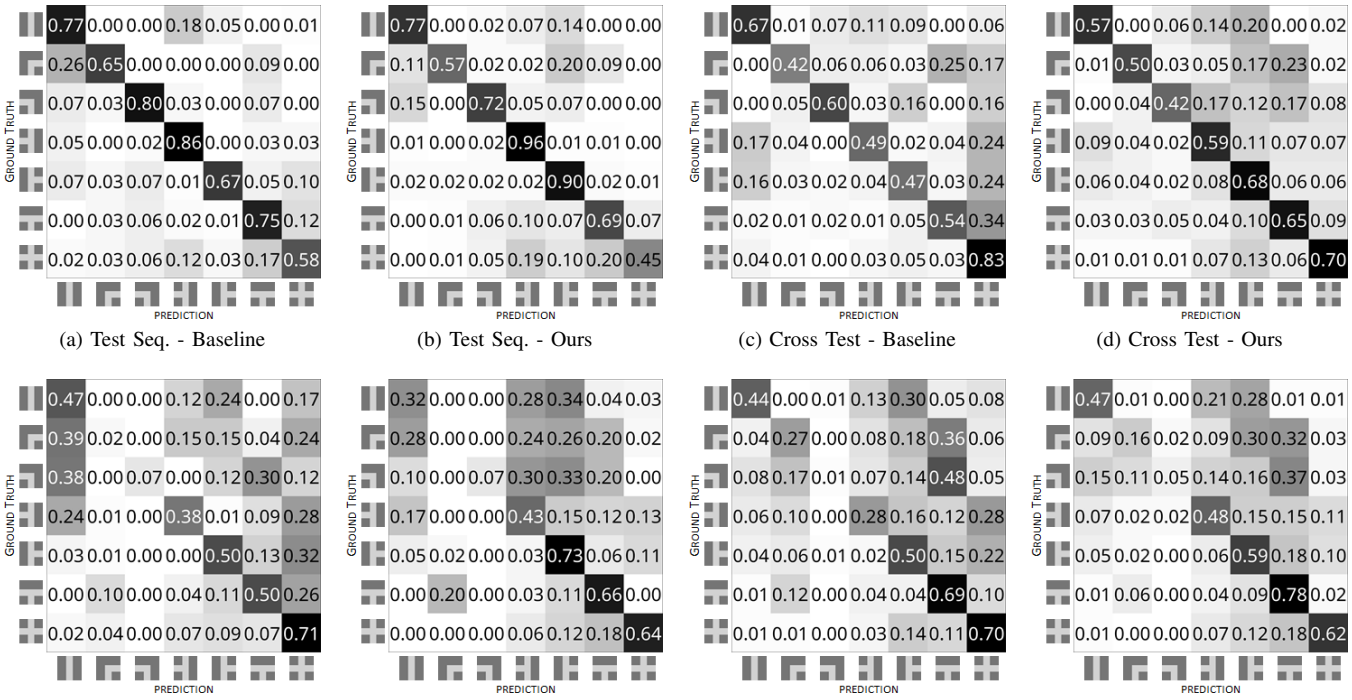
To evaluate the classification performances of our approach, we used the following data.

1) KITTI: We used the work in [6] to select *cam_02—03* color images, raw LiDAR readings and GPS-RTK positions of 8 residential sequences, six recorded on 2011/09/30 [18,20,27,28,33,34] and two recorded on 2011/10/03 [27,34]. Frames were automatically chosen from the whole sequence by gathering only those that are close up-to $20m$ from the intersection center. We refer the reader to the original publication for further details. The major issue with this dataset lies in the relatively low number of intersections and the strong imbalance, see Table II. Considering it would be desirable that all dataset splits, *i.e.,* training, validation, and testing, have all types of intersections, the lack of balance forced us to split this dataset only into train and validate splits. Please notice that randomly choosing frames from the whole dataset was not an option. The reason is due to the multiple frames associated with every intersection. By randomly selecting frames, it would have been possible to include clearly similar frames of the same intersection into both training and validation or testing, frustrating the separation efforts. This left us no choice but to train/validate on this dataset and test on KITTI-360.

2) KITTI-360: This dataset contains ten new sequences recorded in 2013, almost two years after the first recordings. Unfortunately, at the moment, no global positioning information is still provided. We then manually labeled the images, including only those images clearly containing intersections, using the images from the previous dataset as a visual-guide. This dataset presents a much more balanced set of intersections, allowing us to create good dataset splits and test the previous KITTI dataset.

B. Evaluation method and Results

Regarding the evaluation of the obtained classification, we first created an extensive baseline using the Pytorch implementations of RESNET-18 [25] and VGG-11 [26] networks, both with standard Cross Entropy loss (CE) and Focal loss (FC) [27] to evaluate different performances. From the original KITTI dataset, for each intersection, we select all the frames closer than $20m$ to the intersection center, and selected similar appearing frames from KITTI-360 as no geo-referenced position is still available. As opposed to what has been previously done in the works presented in



(a) Test Seq. - Baseline (b) Test Seq. - Ours (c) Cross Test - Baseline (d) Cross Test - Ours

(e) Test Seq. - Baseline (f) Test Seq. - Ours (g) Cross Test - Baseline (h) Cross Test - Ours

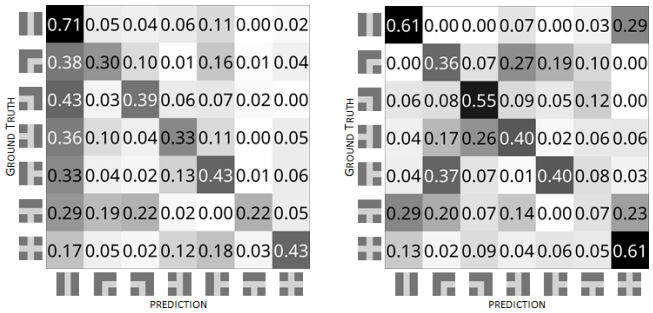
Fig. 6: Confusion matrices: first row, training on KITTI-360; second row, training on KITTI. *Test Seq* matrices refer to test executed on the sequence test of same dataset, *Cross Test* refers to experiments with train/test executed on opposite datasets.

Section I, we performed a per-frame classification aiming at evaluating the abstraction capabilities of modern DNNs. For this reason, the most suitable comparisons concerning the work in [6] are those that consider sequences starting from $20m$ up to the intersection, see Figure 7a. A second comparison can be made with the results of [17]. In both cases, our approach improved the performances of these contributions. Results shown in Table I and Figure 6 clearly show that our system obtained better results using KITTI-360. The lower performances obtained with KITTI might be explained by its strong unbalance, in particular as regards intersection classes 1 and 2. Comparing our results to [6], under the most similar conditions shown in Figures 6f and 7a, we can see that our work achieved better results in most classes, except for classes 1 and 2. However, the good performances achieved using the KITTI-360 even in cross-dataset experiments support our intuitions on the poor balancing of KITTI sequences, see Figures 6b and 6d. The baseline study involved RGB images as well as 2D image-warpings obtained with fixed plane homography. Experiments in Table I show that the classification obtained using warpings instead of *direct* RGB images obtained better results with both tested backbones, see Figure 6. This led us to state that the classification is better performed using this viewpoint instead of the classic perspective, supporting the model-based learning proposed in this work. Finally, even though FC outperformed CE on RGB images, no significant improvements were achieved using warping images. Regarding the teacher/student learning paradigm, the experimental activity reported in Table I, including testing on both KITTI and KITTI-360 with RESNET-18 and VGG networks and different input data including RGB, MBEVs, 3DMASKED-BEVs, 3D-BEVs and WARPINGs images, matches or exceed the results of the comparable cases.

TABLE II: Intersections per-class on evaluated datasets

Sequence	0	1	2	3	4	5	6
2011_09_30_drive_0018	34	✗	✗	41	23	60	247
2011_09_30_drive_0020	21	✗	✗	✗	45	✗	18
2011_09_30_drive_0027	✗	✗	✗	25	17	20	152
2011_09_30_drive_0028	75	51	19	44	110	131	197
2011_09_30_drive_0033	49	39	✗	62	17	16	19
2011_09_30_drive_0034	15	✗	✗	77	24	26	✗
2011_10_03_drive_0027	19	✗	25	139	90	183	217
2011_10_03_drive_0034	46	49	82	70	113	64	84
Total	259	139	126	458	439	500	934
2013_05_28_drive_0000	133	46	40	204	153	147	196
2013_05_28_drive_0002	4	664	679	200	93	321	93
2013_05_28_drive_0003	✗	31	✗	✗	✗	27	✗
2013_05_28_drive_0004	379	154	205	128	125	169	109
2013_05_28_drive_0005	69	31	36	76	153	101	✗
2013_05_28_drive_0006	14	34	66	116	66	132	100
2013_05_28_drive_0007	158	✗	12	42	11	✗	8
2013_05_28_drive_0009	318	70	106	305	111	276	622
2013_05_28_drive_0010	34	14	59	31	29	✗	38
Total	1109	1044	1203	1102	741	1173	1166

Frame numbers of KITTI and KITTI-360 respectively. Regarding KITTI, seq. 2011_09_30_drive_0028 was included in the training set, whilst seq. 2011_10_03_drive_0034 was used in the validation phase and KITTI-360 was used for testing purposes.



(a) Results in [6] (b) Results in [17]

Fig. 7: Confusion matrices of the most similar state of the art classifiers.

V. Conclusions

The work presented proposes a comparison of *direct* DNN-based intersection classifiers along with an evaluation of the teacher/student training paradigm. An extensive experimental activity shows that DNN outperforms previous approaches even without any temporal integration. We demonstrated that the teacher/student allows for a more reliable intersection classification on KITTI datasets. As the benefits of temporal integration are undisputed, we envision developing a system to integrate the results of this research as part of our future work.

References

[1] "Fatality and Injury Reporting System Tool of U.S. National Highway Traffic Safety Administration," https://cdan.dot.gov/query, accessed: 2021-05-21.

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[5] M. Menze, C. Heipke, and A. Geiger, "Joint 3d estimation of vehicles and scene flow," in *ISPRS Workshop on Image Sequence Analysis (ISA)*, 2015.

[6] A. L. Ballardini, D. Cattaneo, S. Fontana, and D. G. Sorrenti, "An online probabilistic road intersection detector," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2017.

[7] A. L. Ballardini, D. Cattaneo, and D. G. Sorrenti, "Visual localization at intersections with digital maps," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2019.

[8] A. Hernández, S. Woo, H. Corrales, I. Parra, E. Kim, D. F. Llorca, and M. A. Sotelo, "3d-deep: 3-dimensional deep-learning based on elevation patterns for road scene interpretation," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 892–898.

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[10] J. Xie, M. Kiefel, M.-T. Sun, and A. Geiger, "Semantic instance annotation of street scenes by 3d to 2d label transfer," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[11] Y. Ki and D. Lee, "A traffic accident recording and reporting model at intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 188–194, 2007.

[12] G. Golembiewski, B. E. Chandler *et al.*, "Intersection safety: A manual for local rural road owners," United States. Federal Highway Administration. Office of Safety, Tech. Rep., 2011.

[13] T. R. Kushner and S. Puri, "Progress in road intersection detection for autonomous vehicle navigation," in *Mobile Robots II*, vol. 852. International Society for Optics and Photonics, 1987, pp. 19–24.

[14] A. J., C. B., S. K.-B., and E. Kim, "Novel intersection type recognition for autonomous vehicles using a multi-layer laser scanner," *Sensors*, 2016.

[15] D. Habermann, C. E. O. Vido, F. S. Osório, and F. Ramos, "Road junction detection from 3d point clouds," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 4934–4940.

[16] D. Bhatt, D. Sodhi, A. Pal, V. Balasubramanian, and M. Krishna, "Have i reached the intersection: A deep learning-based approach for intersection detection from monocular cameras," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4495–4500.

[17] T. Koji and T. Kanji, "Deep intersection classification using first and third person views," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 454–459.

[18] F. Yan, K. Wang, B. Zou, L. Tang, W. Li, and C. Lv, "Lidar-based multi-task road perception network for autonomous vehicles," *IEEE Access*, vol. 8, pp. 86 753–86 764, 2020.

[19] U. Baumann, Y. Huang, C. Gläser, M. Herman, H. Banzhaf, and J. M. Zöllner, "Classifying road intersections using transfer-learning on a deep neural network," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 683–690.

[20] A. L. Ballardini and D. Cattaneo. KITTI Intersection Ground Truth. (2020, Oct 30). [Online]. Available: https://ira.disco.unimib.it/an-online-probabilistic-road-intersection-detector

[21] H. Xu and J. Zhang, "Aanet: Adaptive aggregation network for efficient stereo matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1959–1968.

[22] D. Cattaneo, M. Vaghi, S. Fontana, A. L. Ballardini, and D. G. Sorrenti, "Global visual localization in lidar-maps through shared 2d-3d embedding space," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4365–4371.

[23] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.

[24] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Advances in Neural Information Processing Systems, Autodiff Workshop*, 2017.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[27] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.