

SADA: Unsupervised Domain Adaptation for Reliable Scene Awareness

Hossein Maghsoumi¹, Yaser P. Fallah¹, George Atia^{1,2}

¹*Department of Electrical and Computer Engineering*

²*Department of Computer Science*

University of Central Florida, Orlando, USA

Emails: {hossein.maghsoumi, yaser.fallah, george.atia}@ucf.edu

Abstract—Although deep neural networks are extensively employed in autonomous driving, they must be adjusted to handle unforeseen environmental conditions that were not included in their initial training. Unsupervised domain adaptation (UDA) techniques have proven effective in transferring learned knowledge from one domain to another when the labels in the target domain are unavailable. However, their potential application in scene awareness remains underexplored. This paper introduces SADA (Scene Awareness with Unsupervised Domain Adaptation), a novel approach designed to enhance the reliability of drivable area and lane detection tasks in autonomous driving systems. We leverage UDA techniques in unified multi-task networks to address domain shift challenges, ensuring robust feature extraction under diverse and complex conditions. The proposed technique is evaluated on different real-world datasets and the results demonstrate that the proposed methods outperform the baseline scheme in terms of detection accuracy and consistency.

Index Terms—Scene Awareness, Unsupervised Domain Adaptation, Multi-Task Networks

I. INTRODUCTION

The identification of drivable areas and lane detection are key components of camera-based perception systems for autonomous driving. They form the basis for path planning decisions in autonomous vehicles (AV) and are integral to various advanced driver assistance systems (ADAS) modules, such as lane departure warning (LDW), lane change warning (LCW), and lane-keeping assistance (LKA). Due to safety and efficiency concerns, these systems must demonstrate high stability and precision in diverse and challenging environments. For example, in harsh weather or complex lighting conditions, vehicles need to accurately identify roads and lanes for precise path planning and decision-making [1]. However, when models are trained under specific conditions and tested under different ones, their performance can degrade significantly. This degradation is due to data bias between training and test datasets, including variations in scene appearance, spatial layout, and other real-world discrepancies, commonly referred to as domain shift [2].

The current state-of-the-art research in drivable area and lane detection primarily relies on open-source datasets [3]. Developing a robust model requires training on extensive datasets that cover diverse scenarios and environmental conditions. However, labeling these datasets is expensive and time-consuming, which poses a significant challenge.

Unsupervised Domain Adaptation (UDA) techniques offer a promising solution to this challenge. UDA leverages knowledge from a source domain to enhance the model learning process in a different, yet related, unlabeled target domain. By incorporating information from source domain datasets, UDA helps bridge the domain gap, facilitating more robust learning in the target domain.

While domain adaptation (DA) is well-studied in image segmentation [4], object detection [5], object classification [6], and digit recognition [7], its application in drivable area and lane detection for autonomous driving remains understudied. Recently, unified multi-task networks for the simultaneous detection of drivable areas and lane lines have shown high performance, addressing real-time requirements. Our main contribution in this paper is to develop a solution for robust drivable area and lane detection in unknown, unlabeled domains of driving images by leveraging UDA techniques in unified multi-task networks. We collected an unlabeled target dataset from diverse weather and illumination conditions and used it in our target domain training, along with labeled open-source datasets for knowledge transfer. The following summarizes the main contributions of our work.

- We introduce UDA techniques into multi-task networks to achieve simultaneous drivable area and lane detection addressing the challenges posed by domain shift.
- We develop an adversarial approach to facilitate the learning of domain-invariant features in the underlying multi-task architecture.
- We fuse information from diverse images, which enhances our model’s ability to discern the distributional similarity between source and target images.
- We conduct a comprehensive evaluation on numerous datasets with various illumination and weather conditions, to assess the robustness and performance of our proposed methodology.

II. RELATED WORK

A. Segmentation on Drivable Area and Lane Detection

Semantic segmentation involves assigning tags at the pixel level in images. Deep neural network-based segmentation models have achieved significant success recently and are extensively used in tasks such as drivable area and lane detection for autonomous driving. The authors in [8] propose a model

that blends shallow and deep features within a neural network, enhancing the acquisition of spatial and semantic feature maps with increased accuracy. Reference [9] introduces a unified neural network that establishes context tensors between sub-task decoders instead of segregating tasks within the decoder, facilitating the sharing of designated influence among tasks. [10] employs the spatial pyramid pooling (SPP) module [11] during feature extraction to merge features at various scales, and leverage the Feature Pyramid Network (FPN) module [12] to integrate features across different semantic levels. [13] proposes an effective end-to-end perception network with automatic aspect ratios for each level within the weighted bidirectional feature network and a streamlined training loss function and strategy to enhance accuracy and performance. However, achieving optimal performance requires a carefully labeled dataset, and the labeling process for each new scene incurs additional expenses in time and human resources. To reduce the burden of manual labeling and minimize re-training costs, addressing domain discrepancy is crucial. This ensures that models trained on a labeled dataset perform comparably in another domain even without annotations.

B. Unsupervised Domain Adaptation

UDA techniques aim to align feature distributions between source and target domains. Recent deep UDA methods focus on acquiring domain-invariant features [3] and can be categorized as adversarial discriminative, discrepancy-based, adversarial generative, self-supervision-based, or hybrids. Discrepancy-based methods, such as those in [14] and [6], reduce differences between prediction and activation layers of source and target streams using a loss function. Adversarial discriminative techniques, like the strategy in [4], employ a discriminator to align feature distributions. Adversarial generative techniques integrate a GAN with discriminators at the image and/or feature levels; [15] uses CycleGAN for semantic segmentation adaptation with feature and image discriminators and ensures cycle consistency. Self-supervision-based approaches, such as predicting image rotation in [16], introduce additional self-supervised learning tasks to address the domain gap. While discrepancy-based and self-supervision-based methods are easier to optimize, adversarial learning-based approaches are more effective for demanding tasks like object detection and semantic segmentation due to their robust local feature alignment capabilities. The application of UDA techniques to unified drivable area and lane detection remains unexplored in prior studies.

III. METHOD

A Scene Awareness network that utilizes a unified architecture for the simultaneous identification of drivable areas and lane lines was introduced in [8], demonstrating superior performance compared to other state-of-the-art methods. It comprises one encoder and two decoders for identifying drivable areas and lane lines. In the context of our study, we harness the power of a Scene Awareness network for simultaneous detection of drivable areas and lane lines when

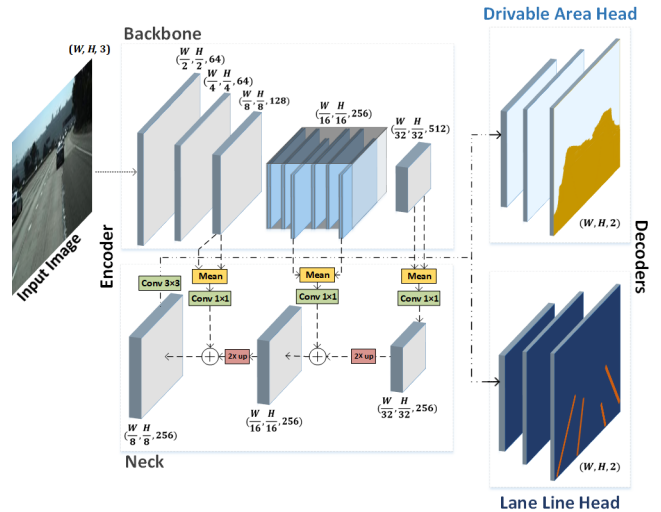


Figure 1. Scene Awareness Architecture.

faced with domain shift challenges. Fig. 1 provides a block diagram illustrating the architecture of the Scene Awareness network used in our approach. In the neck part of this network, it combines the middle layer and the last layer of the final three steps from the backbone part using the mean operator. This network fuses deep semantic features with valuable spatial features in multiple layers at various levels. To mitigate the domain shift, our work develops a Scene Awareness network with UDA capability, which, to the best of our knowledge, is done here for the first time. We leverage an architecture that is augmented with Adversarial Discriminative Domain Adaptation (ADDA) for robust feature extraction under diverse and complex conditions, ensuring the system’s reliability. The next sections provide details of the approach proposed.

A. Basic Adversarial Domain Adaptation

Consider a labeled source dataset (X_s, Y) , where X_s represents the data points along with their labels Y . Each data point $x_s \in X_s \subseteq \mathbb{R}^N$ is associated with a label $y \in [K] := \{1, \dots, K\}$, representing one of K classes. Additionally, there exists an unlabeled target dataset X_t comprising data points $x_t \in \mathbb{R}^N$.

For classification, DA involves two steps: feature extraction and classification. A mapping $M_s : \mathbb{R}^N \rightarrow \mathbb{R}^F$, parameterized by θ_s , extracts $F < N$ features. A classifier $C : \mathbb{R}^F \rightarrow \Delta^K$, parameterized by ϕ , maps features to class probabilities, where Δ^K denotes the probability simplex in K dimensions. For a given input x , the predicted label $p(x)$ is determined by selecting the index that maximizes the probability of class probability, i.e.,

$$p(x) = \operatorname{argmax}_{y \in [K]} C(M_s(x))_y. \quad (1)$$

An effective approach to ADDA was introduced in [17]. In the pre-training phase, both M_s and C are trained on the source

dataset via supervised learning. This optimization is achieved by minimizing the cross-entropy loss as

$$\min_{\theta_s, \phi} \mathbb{E}_{(x_s, y) \sim (X_s, Y)} [-\log(C(M_s(x)))_y]. \quad (2)$$

For simplicity of notation, we omit explicit dependence on the parameters.

In the subsequent adversarial adaptation phase, DA techniques are applied to both the source and target features. A target feature extractor $M_t: \mathbb{R}^N \rightarrow \mathbb{R}^F$, parametrized by θ_t , and discriminator $D: \mathbb{R}^F \rightarrow \mathbb{R}$, parametrized by ψ , are introduced. These components are trained using the target dataset X_t , source data from X_s , and the feature extractor M_s . The discriminator is trained through the optimization objectives defined in (3) and (4) using alternating minimization.

$$\min_{\psi} -\mathbb{E}_{x_s \sim X_s} \log[D(M_s(x_s))] - \mathbb{E}_{x_t \sim X_t} \log[1 - D(M_t(x_t))] \quad (3)$$

$$\min_{\theta_t} -\mathbb{E}_{x_t \sim X_t} \log[D(M_t(x_t))] \quad (4)$$

During inference, M_t and C are used to predict a label for input x as

$$p(x) = \operatorname{argmax}_{y \in [K]} C(M_t(x))_y. \quad (5)$$

B. ADDA for Scene Awareness

The Scene Awareness network in our approach comprises two principal stages: feature extraction (encoder) and head (decoder). The encoder stage is characterized by a function $E_s: \mathbb{R}^N \rightarrow \mathbb{R}^F$, parameterized by θ_s , which extracts a set of F features from the input data x . In the decoder stage, we utilize two mapping heads, $H_i: \mathbb{R}^F \rightarrow [0, 1]^N$, $i \in \{1, 2\}$, parameterized by ϕ_i . The mapping head H_1 maps the features of a data frame to the output of the drivable area head, with entries H_1^n representing the probability for each pixel $n \in \{1, \dots, N\}$ to be foreground or background in the drivable area detection task. Similarly, H_2 maps the features of a data frame to the lane line head, with entries H_2^n representing the probability for pixel n to be foreground or background in the lane line detection task. The predicted drivable area frame is then obtained by selecting the pixels with a probability $H_1^n(E_s(x)) > 0.5$ as foreground, while the lane line frame is determined by considering pixels with a probability $H_2^n(E_s(x)) > 0.5$ as foreground.

Fig. 2 represents the different steps of the proposed algorithm for Scene Awareness Domain Adaptation (SADA). We have a labeled source dataset, denoted as (X_s, Y_1, Y_2) , where X_s is a collection of data frames. Each data frame $x_s \in X_s \subseteq \mathbb{R}^N$ is assigned both a frame label $y_1 \in Y_1 \subseteq \{-1, 1\}^N$ for drivable area detection and $y_2 \in Y_2 \subseteq \{-1, 1\}^N$ for lane line detection tasks. Additionally, we have an unlabeled target dataset, X_t , containing data points $x_t \in \mathbb{R}^N$. The SADA algorithm consists of three steps, which we elaborate on next.

Step 1: Pre-training: In this initial step, we train the encoder function E_s and two head functions H_1 and H_2 using supervised learning on the source data. The optimization process

aims to find the optimal parameters θ_s and ϕ , by employing the cross-entropy loss and representing labels in a one-hot encoding format. We define the total loss, denoted as $Loss$, as a weighted combination of the binary classification losses for drivable area (L_{da}) and lane line (L_{ll}) tasks as

$$Loss = k_1 \sum_i^N L_{da}(H_1^i, y_1^n) + k_2 \sum_n^N L_{ll}(H_2^n, y_2^n). \quad (6)$$

Here, L_{da} is the binary classification loss function, defined in (7) based on the cross-entropy, serving as the loss for drivable area segmentation. The loss L_{ll} is the lane line classification loss function defined in (8), in which we utilize focal loss (FL) involving a modulating term to the cross-entropy, to emphasize learning on challenging, misclassified examples [18].

$$L_{da}(H_1^n, y_1^n) = \begin{cases} -\log(H_1^n) & \text{if } y_1^n = 1 \\ -\log(1 - H_1^n) & \text{otherwise} \end{cases} \quad (7)$$

$$L_{ll}(H_2^n, y_2^n) = \begin{cases} -(1 - H_2^n)^\gamma \log(H_2^n) & \text{if } y_2^n = 1 \\ -(H_2^n)^\gamma \log(1 - H_2^n) & \text{otherwise} \end{cases} \quad (8)$$

In (8), γ is the focusing parameter set to 2, and $y_i^n \in \{\pm 1\}$ represents the groundtruth for head i , while $H_i^n \in [0, 1]$ is the predicted probability for the label $y_i^n = 1$. The weights k_1 and k_2 in the total loss function can be adjusted to balance the contributions of the drivable area and lane line tasks. This training process, producing E_s, H_1 and H_2 , is summarized by the following function:

$$\{E_s, H_1, H_2\} = \text{StandardTraining}(X_s, Y_1, Y_2). \quad (9)$$

Step 2: Adversarial Adaptation in Scene Awareness: Having obtained the source encoder function and head functions, in the next step we use an adversarial adaptation function to improve the semantic segmentation. Two additional functions are introduced: the target encoder function denoted as $E_t: \mathbb{R}^N \rightarrow \mathbb{R}^F$, parameterized by θ_t , and the discriminator function denoted as $D: \mathbb{R}^F \rightarrow \mathbb{R}$, parameterized by ψ . We train these functions using the unlabeled target dataset X_t , source data points from X_s (without labels), and the learned source data encoder function E_s from the initial stage. The discriminator function acts as a binary classifier and is trained by solving the two unconstrained optimization problems in (10) and (11) simultaneously, using cross-entropy losses. The solution is obtained through alternating minimization.

$$\min_{\psi} -\mathbb{E}_{x_s \sim X_s} \log[D(E_s(x_s))] - \mathbb{E}_{x_t \sim X_t} \log[1 - D(E_t(x_t))] \quad (10)$$

$$\min_{\theta_t} -\mathbb{E}_{x_t \sim X_t} \log[D(E_t(x_t))] \quad (11)$$

In summary, the adversarial adaptation procedure utilizes X_s, X_t , and E_s from (9) to derive E_t , expressed using shorthand notation as

$$E_t = \text{AdversarialAdaptation}(X_s, X_t, E_s). \quad (12)$$

Step 3: Inference time: Finally, the SADA network is constructed with source decoders H_1 and H_2 from the pre-training

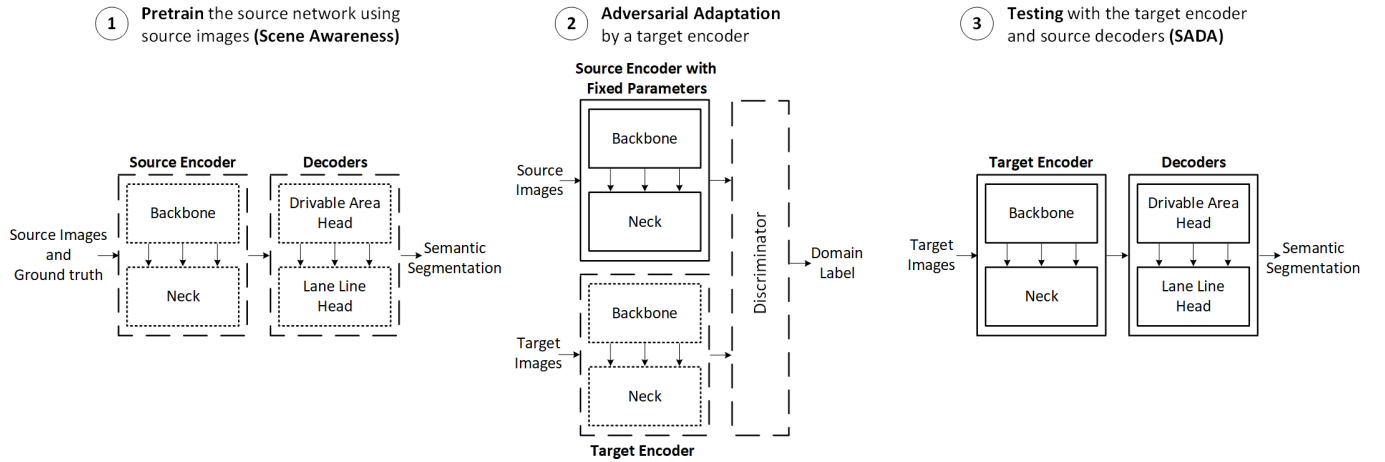


Figure 2. A block diagram of SADA Network.

step and the target encoder function E_t trained with ADDA. During inference, SADA predicts the drivable frame region by selecting the pixels with a probability $H_1^n(E_t(x)) > 0.5$ and lane line frame region by selecting the pixels with a probability $H_2^n(E_t(x)) > 0.5$.

IV. EXPERIMENTS

A. Implementation Details

The Scene Awareness network is pre-trained using the BDD100k dataset’s training set with specific scene labels. We built the network using PyTorch, setting the initial learning rate at 0.001 and employing the Adam optimizer with polynomial decay (exponent 0.9) to improve learning efficiency. The batch size is set to 8. Experiments were conducted on a machine with an Intel(R) Core(TM) i9-9900KF CPU at 3.60 GHz and 32 GB of RAM.

B. Datasets

We selected the DSDLDE, JuJu, and Tusimple datasets for lane detection due to their challenging scenarios [19]–[21] featuring harsh weather, challenging illumination, and diverse road conditions. For visual assessment, we utilized a test set of 200 images from DSDLDE, focusing on the most challenging scenes. Our quantitative evaluation involved the Mapillary, CamVid, KITTI, and Cityscapes datasets. The DSDLDE dataset comprises approximately 92,000 images with harsh weather and lighting conditions. JuJu includes 2,045 frames of highways and secondary roads under challenging illumination, weather, and traffic conditions. Tusimple contains 6,408 sequences of highway scenes featuring various instances of poor-quality roads, worn lanes, and challenges posed by shadows. The Mapillary dataset consists of 25,000 images labeled for various weather conditions, times of day, and camera viewpoints. CamVid captures driving scenes through a dashboard camera with over 700 images annotated in 32 classes. KITTI is popular for autonomous driving research, focusing on traffic scenes with drivable areas and lane lines grouped under a single ‘Road’ label. Similarly, Cityscapes is a

large-scale urban scene understanding database, with drivable areas and lane lines grouped under a ‘Road’ category.

C. Evaluation Metrics

We assess the source and target networks by comparing their road and lane line detections to the groundtruth using the Intersection over Union (IoU) and Mean Intersection over Union (MIoU) metrics for semantic segmentation. The IoU is defined as:

$$IoU = \frac{TP}{TP + FP + FN} \quad (13)$$

where TP , FP , and FN represent the true positive, false positive, and false negative rates, respectively. The MIoU, providing a comprehensive assessment, is calculated as:

$$MIoU = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FP_i + FN_i} \quad (14)$$

where N is the number of classes, improving the evaluation of network accuracy in matching the groundtruth.

D. Results

Visual Evaluation: We conduct experiments on frames featuring the most challenging scenarios and present results for both the Scene Awareness Network and the proposed SADA Network. Fig. 3 showcases ten scenes, highlighting various challenges encountered during day and night. The frames are from the DSDLDE dataset, and the subsequent section elaborates on the observed results.

In the first row of Fig. 3, both day and night scenes show a wet road with rain marks on the car’s windshield. During the day, our model not only demonstrates stronger performance against line fading, but also exhibits higher robustness against reflections within the car’s glass, evident in both road and lane detection tasks. In the night scene, despite low visibility and glare, our model maintains stability in detecting drivable areas and lanes. The second row features snowy scenes, where our model demonstrates robustness on snow-covered roads. In the daytime scene of the third row, alongside challenges

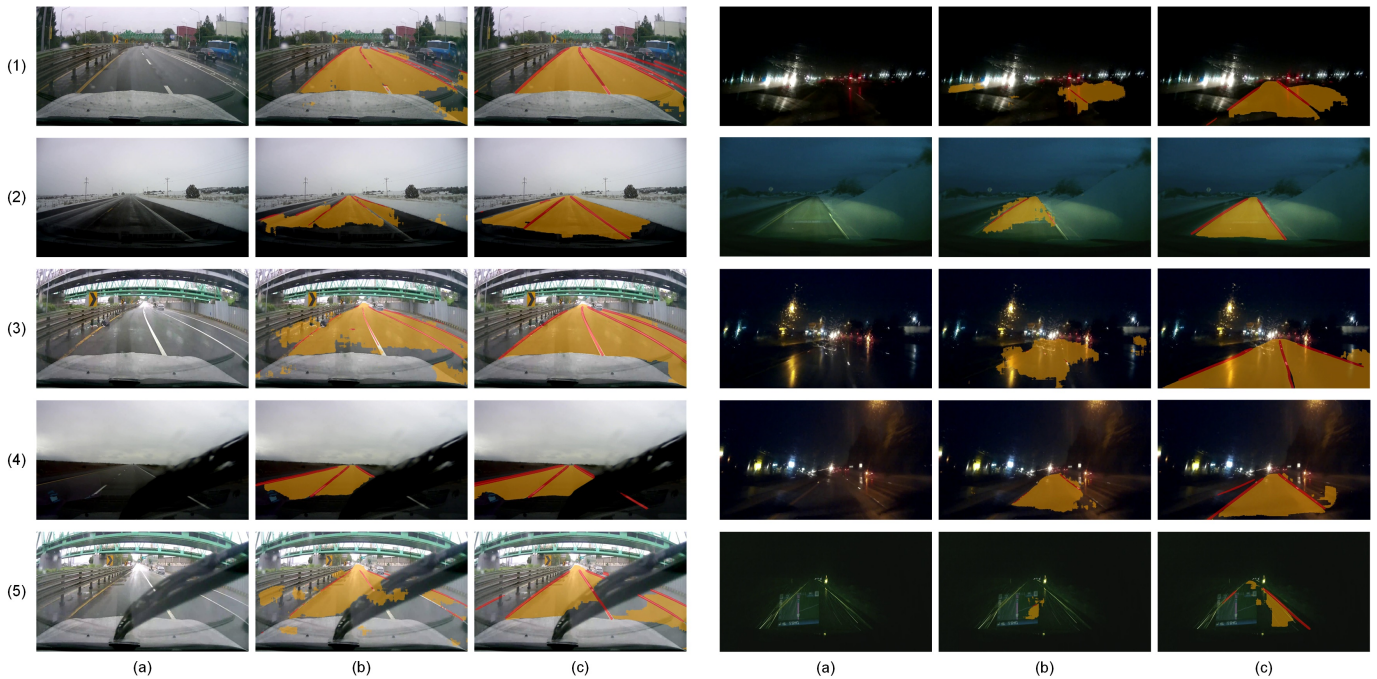


Figure 3. Comparison of Scene Awareness and SADA networks: (a) Original image, (b) Scene Awareness result (c) SADA result.

from the first row, we face issues with worn lines and roadside obstacles. Our model shows fewer false positives and better true positive detection. The nighttime scene from this row demonstrates our model’s notable robustness against wet road reflections and glare. The fourth row depicts rainy weather in both day and night scenes, with wipers and poor visibility presenting challenges. Our model performs better in these conditions. In the daytime scene of the fifth row, a windshield wiper occupies much of the image, but our method handles this well. In the night scene of this row, numerous challenges persist, including snowfall, high reflection in the windshield, and low visibility. While our method may not have achieved precise segmentation of the drivable area and lane, it demonstrates a superior capability compared to the previous method.

Quantitative Evaluation: Table I summarizes the results on the Mapillary dataset. For lane line detection, the SADA network achieves a higher IoU of 28.66% compared to the Scene Awareness Network’s 27.79%, indicating more accurate lane line localization. For drivable area detection, the SADA Network also outperforms the Scene Awareness Network with an IoU of 90.54% versus 89.21%, demonstrating superior spatial understanding and precision. The MIoU values, which represent the average IoU across all classes, further emphasize the overall performance improvement of SADA (59.6%) over Scene Awareness (58.5%).

Table II presents the results on the CamVid dataset. The SADA Network again shows higher IoU for lane line detection at 27.31% compared to the Scene Awareness Network’s 26.92%, maintaining more accurate localization. For drivable area detection, the SADA Network outperforms the

Table I
QUANTITATIVE RESULT ON MAPILLARY DATASET

Network	Drivable Area IoU(%)	Lane Line IoU(%)	All MIoU(%)
Scene Awareness	89.21	27.79	58.5
SADA	90.54	28.66	59.6

Table II
QUANTITATIVE RESULT ON CAMVID DATASET

Network	Drivable Area IoU(%)	Lane Line IoU(%)	All MIoU(%)
Scene Awareness	88.54	26.92	57.73
SADA	90.27	27.31	58.79

Scene Awareness Network with an IoU of 90.27% versus 88.54%, demonstrating consistent superior performance and robust comprehension of road environments across datasets. The MIoU values of 58.79% for SADA and 57.73% for Scene Awareness further confirm the SADA Network’s overall effectiveness.

Table III presents our results on the KITTI and CitySpace Datasets. Since these two datasets do not have separate lane labels, we treated the output of both drivable areas and lane segments as roads and compared them to the groundtruth. Our results show that SADA achieves higher IoU scores compared to the baseline method on both datasets, with 89.03% on KITTI and 88.27% on Cityscapes, surpassing the baseline by 3.11% and 1.86%, respectively.

The comparative performance of the models across datasets

Table III
QUANTITATIVE RESULT ON KITTI & CITYSCAPES DATASETS

Network	KITTI Road IoU(%)	Cityscapes Road IoU(%)
Scene Awareness	85.92	86.41
SADA	89.03	88.27

reveals that the SADA Network consistently achieves higher accuracy in both lane line detection and drivable area detection under diverse scenarios, which are quantitatively significant for this application.

V. CONCLUSION

This paper introduces SADA, a new strategy for enhancing the reliability of drivable area and lane detection in autonomous vehicles. By introducing Unsupervised Domain Adaptation (UDA) within unified multi-task networks, we tackle domain shifts and promote robust feature extraction under varied conditions. The model benefits from integrating diverse images, improving the alignment of distribution between source and target images and enhancing knowledge transfer. Our evaluations demonstrate that SADA outperforms the baseline method across all test datasets.

ACKNOWLEDGMENT

This work was supported by NSF under Award CCF-2106339 and DARPA under Agreement No. HR0011-24-9-0427.

REFERENCES

- [1] H. Maghsoumi, N. Masoumi, and B. Araabi, "Lane detection and tracking datasets: Efficient investigation and new measurement by a novel dataset scenario detector application," *IEEE Trans. on Instrum. and Meas.*, vol. 73, 2024.
- [2] C. Li, B. Zhang, J. Shi, and G. Cheng, "Multi-level domain adaptation for lane detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.
- [3] C. Hu, S. Hudson, M. Ethier, M. Al-Sharman, D. Rayside, and W. Melek, "Sim-to-real domain adaptation for lane detection and classification in autonomous driving," in *IEEE Intelligent Vehicles Symposium (IV)*, 2022.
- [4] M. Wulfmeier, A. Bewley, and I. Posner, "Addressing appearance change in outdoor robotics with adversarial domain adaptation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
- [5] D. Ho, K. Rao, Z. Xu, E. Jang, M. Khansari, and Y. Bai, "Retinagan: An object-aware approach to sim-to-real transfer," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
- [6] C. Chen, Z. Fu, Z. Chen, S. Jin, Z. Cheng, X. Jin, and X. Hua, "Homm: Higher-order moment matching for unsupervised domain adaptation," in *Proc. of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34.
- [7] I. Alkhouri, A. Awad, C. Hatfield, and G. Atia, "A discriminative approach to unsupervised domain adaptation in coarse-to-fine classifiers," in *IEEE Int. Workshop on Machine Learning for Sig. Proc. (MLSP)*, 2023.
- [8] H. Maghsoumi, N. Masoumi, and B. Araabi, "Roadsave: A robust lane detection method based on validity borrowing from reliable lines," *IEEE Sensors Journal*, vol. 23, no. 13, 2023.
- [9] Y. Qian, J. Dolan, and M. Yang, "Dlt-net: Joint detection of drivable areas, lane lines, and traffic objects," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4670–4679, 2019.
- [10] D. Wu, M. Liao, W. Zhang, X. Wang, X. Bai, W. Cheng, and W. Liu, "Yolop: You only look once for panoptic driving perception," *Machine Intelligence Research*, vol. 19, no. 6, pp. 550–562, 2022.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. on Pattern Anal. and Machine Intelligence*, vol. 37, no. 9, 2015.
- [12] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.
- [13] D. Vu, B. Ngo, and H. Phan, "Hybridnets: End-to-end perception network," *preprint arXiv:2203.09035*, 2022.
- [14] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Computer Vision–ECCV Workshops: Amsterdam, The Netherlands, Proceedings Part III 14*. Springer, 2016.
- [15] J. Hoffman, E. Tzeng, T. Park, J. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *International Conference on Machine Learning*, 2018.
- [16] J. Xu, L. Xiao, and A. López, "Self-supervised domain adaptation for computer vision tasks," *IEEE Access*, vol. 7, pp. 156694–156706, 2019.
- [17] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. of the IEEE Conf. on Comp. Vision and Patt. Recog.*, 2017.
- [18] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. of the IEEE Int. Conf. on Computer Vision*, 2017, pp. 2980–2988.
- [19] "DSDLDE v.0.9: Video clips for lane marking detection," https://drive.google.com/file/d/1315Ry7isciL-3nRvU5SCXM/_-4meR2MyI/view, 2017.
- [20] J. Yoo, "Juju1006s," <https://sites.google.com/site/juju1006s/>, 2016.
- [21] K. Zhou, "Tusimple-benchmark," <https://github.com/TuSimple/tusimple-benchmark>, 2017.