# 3DLaneNAS: Neural Architecture Search for Accurate and Light-Weight 3D Lane Detection

Ali Zoljodi[1], Mohammad Loni[1], Sadegh Abadijou[1], Mina Alibeigi[2], and Masoud Daneshtalab[1,3]

[1] School of Innovation, Design and Engineering, Mälardalen University, Sweden
{ali.zoljodi, mohammad.loni, masoud.daneshtalab}@mdu.se
s.abadijou@gmail.com
[2] Zenseact AB, mina.alibeigi@zenseact.com
[3] Department of Computer Systems, Tallinn University of Technology, Estonia

**Abstract.** Lane detection is one of the most fundamental tasks for autonomous driving. It plays a crucial role in the lateral control and the precise localization of autonomous vehicles. Monocular 3D lane detection methods provide state-of-the-art results for estimating the position of lanes in 3D world coordinates using only the information obtained from the front-view camera. Recent advances in Neural Architecture Search (NAS) facilitate automated optimization of various computer vision tasks. NAS can automatically optimize monocular 3D lane detection methods to enhance the extraction and combination of visual features, consequently reducing computation loads and increasing accuracy. This paper proposes 3DLaneNAS, a multi-objective method that enhances the accuracy of monocular 3D lane detection for both short- and long-distance scenarios while at the same time providing a fair amount of hardware acceleration. 3DLaneNAS utilizes a new multi-objective energy function to optimize the architecture of feature extraction and feature fusion modules simultaneously. Moreover, a transfer learning mechanism is used to improve the convergence of the search process. Experimental results reveal that 3DLaneNAS yields a minimum of 5.2% higher accuracy and ≈1.33× lower latency over competing methods on the synthetic-3D-lanes dataset. Code is at `https://github.com/alizoljodi/3DLaneNAS`

**Keywords:** Autonomous Vehicles · 3D Lane Detection · Neural Architecture Search

## 1 Introduction

To operate safely, an autonomous vehicle needs a precise understanding of the road perspective. Lane detection is the task of estimating lane markings' positions, which is a crucial part of road understanding [33]. The majority of prior works focused on improving the lane detection accuracy on 2D images [12,18,36,31,5]. 2D lane detection methods work as follows: first, they perform lane segmentation on a 2D image; then, extracted lanes are mapped on a 3D coordination space. Despite 2D lane detection's simplicity, lanes should be projected into 3D. The detected lanes in the image plane are commonly projected to the 3D world using the flat earth assumption, which can lead to erroneous elevation and lane curvature estimates on hilly, banked, or curving roadways. (up to 14% accuracy loss [10]). Multi-sensor 3D lane detection methods aim to overcome this limitation by exploiting images with 3D shapes obtained from stereo-vision cameras or LiDAR sensors. However, multi-sensor 3D lane detection methods suffer from (i) expensive sensor configuration and (ii) erroneous prediction over long distances. Monocular 3D lane detection methods have been proposed to tackle these challenges [10,11]. The monocular 3D lane detection methods work as follows: first, they extract scene features with various scales from a 2D image; then, the extracted features are transformed into a top-view (bird's eye view) since lanes are parallel in the top-view, which helps in their detection precision. Next, using Convolutional Neural Networks (CNNs), top-view features are fused to construct a unified representation. Note that CNNs are known to provide the best results for accurate lane detection [10,11,33]. Finally, the fused features are classified using the fully-connected layer(s) to detect lane markings' positions. Fig. 1 shows the overview of state-of-the-art monocular 3D lane detection methods. The monocular 3D lane detection methods benefit from (i) a cost-efficient sensor configuration and (ii) accurate estimations over long distances compared to multi-sensor 3D lane detection methods (Section 5). Despite the success of monocular 3D lane detection methods, they suffer from inefficient feature extraction and feature fusion modules causing inaccurate predictions.

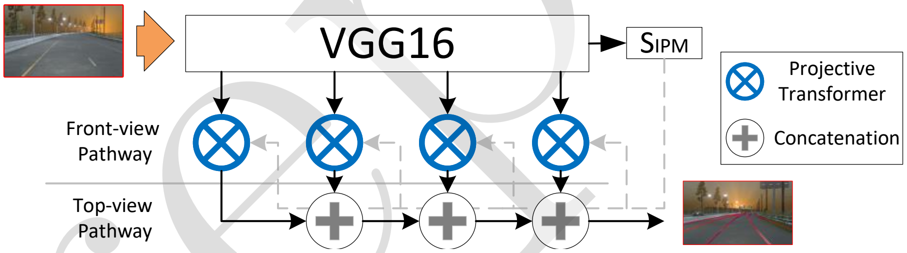


Fig. 1: The overview of the state-of-the-art 3DLaneNet [10] architecture. The feature extraction module is fed a front-view image. The VGGNet-16 [32] architecture is used extract features. Using the sampling information provided by $S_{IPM}$, the projective transformation layers project features to the top-view. To estimate 3D lane coordinations, top-view features are concatenated and processed via a fully-connected network.

Neural Architecture Search (NAS) advances the design procedure in CNNs [21,25,24]. Inspired by NAS's success, we designed a high-performance CNN architecture for monocular 3D lane detection. To this aim, we propose 3DLaneNAS, a multi-objective NAS method that designs an accurate yet efficient monocular 3D lane detection architecture. 3DLaneNAS uses multi-objective simulated annealing (MOSA) to explore the search space since it quickly finds the optimal solution [25]. While state-of-the-art methods provide excellent results in detecting short-distance lanes, our analysis indicates that they are not sufficient for detecting long-distance objects (Section 5). 3DLaneNAS considers short- and long-distance errors and the network inference time in its objective function. 3DLaneNAS also devises a transfer learning mechanism to expedite the search procedure in a large search space.

According to our experiments on the synthetic-3D-lanes dataset [10], 3DLaneNAS outperforms the state-of-the-art monocular 3D lane detection methods by achieving up to 17.5% higher accuracy and 1.2× lower inference time on NVIDIA® RTX A4000. 3DLaneNAS generates similar results with 2.74% standard deviation demonstrating our results are reproducible. To the best of our knowledge, 3DLaneNAS is the first attempt in the literature that successfully develops a NAS method for the 3D lane detection task.

## 2 Related Work

### 2.1 Lane Detection

Convolutional Neural Networks (CNNs) provide the most accurate results for the lane detection task [12,31,5,3,17,28,29,2,16,10,11]. Earlier studies attempted to improve the accuracy of CNN-based lane detection on 2D images. 2D lane detection methods are based on either image classification [12] or image segmentation [31,5]. Some studies on 2D lane detection methods transform front-view to top-view as the post-processing module to provide a precise perception of 3D coordination space [3,17]. A group of researchers proposed that 3D lane detection be accomplished with the use of stereo-vision cameras [28,29]. Although stereo-vision cameras provide a better view of the 3D world, they suffer from accuracy loss in long-distance scenarios. Another group of researchers [2,16] investigated 3D lane detection using multi-sensor techniques. In these methods, lanes' 3D coordinates are estimated by fusing Lidar sensor data and RGB camera. These methods are still costly regarding the sensor configuration.

Recently, a group of studies [10,11] proposed estimating the accurate position of lane markings in 3D world coordinate by utilizing only one front-looking monocular camera. 3DLaneNet [10] is the first effort in the literature proposing intra-network inverse-perspective mapping (IPM) [27] to remove road geometry assumptions. The architecture of 3DLaneNet is shown in Fig. 1. The IPM is front-view to top-view transformation with anisotropic scaling. The transformation is applied by utilizing a fixed set of parameters that specify top-view region boundaries and anisotropic scaling. The transformation parameters are estimated by an additional head directly connected to the feature extraction module. Inspired by [15], $S_{IPM}$ samples the front-view image pixels that aim to assign a new position in the top-view image. To improve the interaction between CNN feature extraction and feature map transformation, 3DLaneNet designed a dual-pathway architecture. 3DLaneNet has improved the lane fitting accuracy by integrating the transformation mechanism in the lane detection architecture. 3DLaneNet leveraged VGGNet-16 [32] pre-trained on the ImageNet dataset [6].

Despite the success of 3DLaneNet, it suffers from (i) inefficient feature extraction/fusion modules and (ii) bounded coordination system used to represent anchors. GenLaneNet [11] proposed an improved anchor representation by decoupling the learning of image encoding and 3D geometry reasoning. However, GenLaneNet still suffers from requiring significant computing resources for the learning process (up to 3M parameters) and inaccurate long-distance estimations.

### 2.2 Neural Architecture Search

Automated machine learning (AutoML) advances the capability of intelligent systems by tweaking hyper-parameters of learning models [13]. Neural Architecture Search (NAS), as a subset of AutoML, aims to design efficient neural networks for complex learning tasks [8]. Early proposed NAS methods employed Reinforcement Learning (RL) [37,14] or evolutionary-based algorithms [23,26] to search through the search space. However, these methods require remarkable computing capacity, for example, 500 NVIDIA® P100 GPUs to evaluate 20,000 neural architectures over four days [38]. Recently, differentiable NAS methods provide state-of-the-art results for various learning tasks [20,21,22]. DARTS [21] is a differentiable NAS method that uses the gradient descent algorithm to search and train neural architecture cells jointly. Despite the success of differentiable NAS methods in various domains [22], they suffer from inefficient training due to interfering with the training of different sub-networks each other [4]. Moreover, it has been proved that with equal search spaces and training setups, differentiable NAS methods converge to similar results [7].

Meta-heuristic-based NAS methods [24,35,25] benefit from fast and flexible algorithms to search a discrete search space. FastStereoNet [25] is a state-of-the-art meta-heuristic method that designs an accurate depth estimation pipeline. Inspired by FastStereoNet, we propose 3DLaneNAS for 3D lane detection. 3DLaneNAS advantages from meta-heuristic NAS approaches' quick convergence search. In addition, we decoupled search space into two modules to have a more efficient search. The results (Section 5) show that the 3DLaneNAS method outperforms two of the most well-known 3D lane detection benchmark models, 3DLaneNet [10] and GenLaneNet [11].

## 3 3DLaneNAS

### 3.1 Search Space

The 3DLaneNAS search space contains feature extraction, projective transformation, and feature fusion modules (Fig. 2). First, the feature extraction module extracts feature maps from the front-view image in four resolution scales. Second, the front-view feature maps are passed throw the projective transformation layers to construct the top-view feature maps. The projective transformation layers estimate input picture pixels' new position on the target image via bilinear interpolation. We set the parameters of the projective transformation layers according to [15]. Next, the feature fusion module concatenates top-view feature maps of multiple resolutions into a single feature space. Finally, a fully-connected layer classifies the output of the feature fusion module to estimate lane positions in 3D world coordinate. In this paper, we used NAS to optimize the architecture of feature extraction (Section 3.1) and feature fusion modules (Section 3.1).

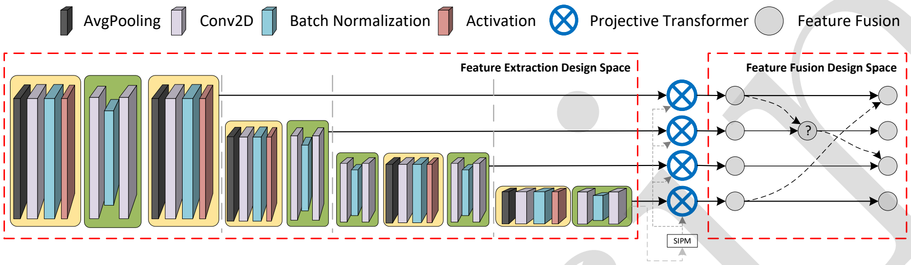
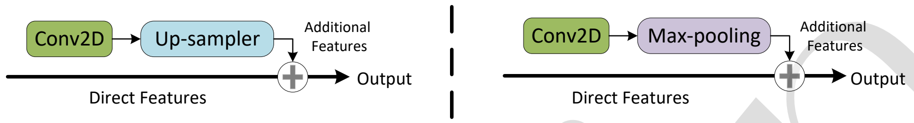


Fig. 2: The overview of 3DLaneNAS architecture. The search space of feature extraction is a stack of `ConvBnAct` and `Squeeze Blocks`. The search space of feature fusion is the combination of projective transformation outputs.

**Feature Extraction Search Space** The feature extraction module computes a hierarchy of feature maps in a certain number of resolutions. Each resolution level has been extracted by a stack of atomic blocks. In this paper, we define two simple atomic blocks, including `ConvBnAct` and `Squeeze Block`. The `ConvBnAct` is a convolution layer followed by batch-normalization and an activation function. `Squeeze Block` works as a light-weight encoder-decoder function to augment the features that have a higher impact on 3D lane detection performance. Each `Squeeze Block` is built by a $1 \times 1$ average-pooling layer followed by two convolution layers with reversed input/output sizes and a batch-normalization in between. Five different output channels are available for `ConvBnAct` and `Squeeze Block`, including 16, 32, 64, 128, 256. We insert or remove an atomic block of feature extraction in each search iteration. The size of the feature extraction search space is proportional to the current state of the feature extraction stack. The minimum size of the feature extraction stack is set to 4 and is limited to 50 blocks. Thus, the maximum size of feature extraction search space is $10^{50}$.

**Feature Fusion Search Space** Fig. 3 represents feature fusion module. The feature fusion module consists of concatenation nodes, where each node can be active or inactive. An active node modifies a feature map by concatenating it with another feature map from a different resolution group. If the feature map should be concatenated with a higher resolution feature map, a max-pooling operation should be used to decrease the additional layer's resolution (down-sampling layer). On the other hand, if the feature map should be concatenated with a lower resolution feature map, an up-sampler operation should be used to increase the additional layer's resolution (up-sampling layer). In each search iteration, one concatenation node swaps from/to active/inactive. The maximum number of active nodes for feature fusion is 20, and the minimum is zero. Each concatenation node can have 12 distinct input combinations. Thus, the size of search space for the feature fusion module is $12 \times 2^{20}$.



(a) Concatenate to lower scale additional features

(b) Concatenate to upper scale additional features

Fig. 3: (a) The Up-sampler is used for concatenating the additional features with lower resolution. (b) The Max-pooling is used for concatenating the additional features with higher resolution.

### 3.2 Search Algorithm

We use multi-objective simulated annealing (MOSA) [1] algorithm to find the near-optimal architecture for 3D lane detection. MOSA's search convergence is faster than genetic programming. [25]. MOSA selects candidates with the probability of $min(1, exp(-\Delta/T))$. $\Delta$ is the difference in energy between the present and the newly generated candidate. $T$ is the regulating parameter for annealing temperature. $T$ starts from a big value ($T_{Max}$) that is gradually decreases to a small value ($T_{Min}$). Early on, $T_{Max}$ must be large enough to choose non-optimal choices. (exploration). On the other hand, $T_{Min}$ should be small enough to only give the maximum selection chance to optimal candidates (exploitation).

We consider a multi-objective energy function (Eq. 1) to improve the 3D lane detection accuracy in addition to reducing the network inference time. The energy function ($E$) is the multiplication of the network inference time ($t$) and the average value of lateral ($LatE$) and longitudinal ($LongE$) errors. $n$ indicates the number of test samples in each batch. We do not use any proxy such as *Floating-Point-Operations-per-Second* (FLOPs) for the inference time estimation. Instead, we run the network directly on the target hardware (NVIDIA® RTX A4000) to measure the exact inference time. We also consider a penalty coefficient ($\alpha$=10) to find the best error-latency trade-off.

$$E = \frac{1}{2n}\sum_{i=1}^{n}((LatE_i + LongE_i)t_i) \times \max(1, \frac{1}{n}\sum_{i=1}^{n}(t_i - \alpha)) \qquad (1)$$

### 3.3 Training Procedure

The training procedure for 3D lane detection is time-consuming ($\approx$ 10 GPU hours for training one candidate). Inspired by [25], we partially train each candidate with fewer epochs to reduce the search time. After achieving 5× reduction search cost, our search process takes ≈ 130 GPU hours on a single NVIDIA® RTX A4000. In addition, 3DLaneNAS leverages the idea of the transferred weights mechanism [25] to expedite the search process.

The loss function is a combination of two equal-weighted terms (Eq. 2): The cross-entropy of lane detection (first term) and least absolute deviations ($\mathcal{L}$1-loss) of predicted lanes and the ground truth.

$$\begin{aligned}\mathcal{L} \quad &= - \sum_{t\in\{c_1,c_2,d\}} \sum_{i=1}^{N} (p_t^i \; log \; p_t^i + (1-\hat{p}_t^i) \; log \; (1-p_t^i)) \\ &+ \sum_{t\in\{c_1,c_2,d\}} \sum_{i=1}^{N} \hat{p}_t^i . (\|x_t^i + \hat{x}_t^i\| + \|z_t^i + \hat{z}_t^i\|)\end{aligned} \tag{2}$$

The $p_t^i$ indicates the confidentiality of detecting the i-th lane section, represented by an anchor. $x$ and $z$ indicate lanes' coordination.

# 4 Experimental Setup

## 4.1 Dataset

To evaluate our proposed method, we use the synthetic-3D-lanes dataset [10]. The dataset has been synthesized by the Blender graphic engine [9]. synthetic-3D-lanes contains more than 300K training samples and 5k validation samples for different illumination and weather conditions. In this dataset, the scene terrains are modeled by a mixture of Gaussian distribution.

## 4.2 Configuration Setup

Hardware specification as well as training and search parameters are summarized in Table 1.

## 4.3 Evaluation Metrics

The performance of lane detection methods is evaluated using the average precision (AP) metric, which is the average percentage of the matched predicted lanes [30]. We also report *lateral*($x$) and *longitudinal*($z$) errors for near (0-40 meters) and far distances (40-100 meters). Additionally, we report the maximum F-score to indicate the application's optimal operation point.

# 5 Experimental Results

## 5.1 3D Lane Detection Performance Metrics

Table 2 presents a comparison of the results obtained by 3DLaneNAS with GenLaneNet [11] and 3DLaneNet [10] as the cutting-edge 3D lane detection methods. The inference time for GPU is measured with batch size set to 1. 3DLaneNAS yields 5.2% and 17.5% higher accuracy compared to GenLaneNet and

Table 1: Summarizing hardware specification, train, and search parameters.

| **Train/Test Hardware Device** | **Specification** |
|---|---|
| GPU | NVIDIA® RTX A4000 |
| GPU Compiler | CUDA v11.3 & cuDNN v8.2.0 |
| DL Framework | PyTorch v1.9.1 |
| **Training and Search Parameters** | **Value** |
| Full-Training Epochs | 30 |
| Search Epochs | 5 |
| Batch Size | 8 |
| Learning Rate | $5 \times 10^{-4}$ |
| Optimizer | Adam |
| $T_{Max}$ / $T_{Min}$ | 2500 / 2.5 |

3DLaneNet, respectively. Compared to GenLaneNet and 3DLaneNet, 3DLaneNAS reduces inference time by 1.33× and 1.2×, respectively. 3DLaneNAS predicts lane positions with 41.9% and 50.4% lower longitudinal error, and 44% and 59% lower lateral error in comparison with GenLaneNet and 3DLaneNet, respectively. Fig. 4 compares the visualization results of 3DLaneNAS with state-of-the-art on three different road scenarios. 3DLaneNAS performs better in curvy, downhill, and uphill road settings with partially visible lanes.

Table 2: Comparing the performance of 3DLaneNAS with the states-of-the-art.

| **Architecture** | **AP** | **F-score** | **Lateral Error (cm)** | | **Longitudinal Error (cm)** | | **#Params** | **Inference** |
|---|---|---|---|---|---|---|---|---|
| | **(%)** | **(%)** | **0-40m** | **40-100m** | **0-40m** | **40-100m** | **(M)** | **Time (ms)** |
| 3DLaneNet [10] | 74.9 | 77.7 | 11.5 | 60.1 | 3.2 | 23.0 | 20.8 | 14.5 |
| GenLaneNet [11] | 87.2 | 83 | 7.4 | 53.8 | 1.5 | 23.2 | 3.36 | 16 |
| 3DLaneNAS | **92.4** | **92.1** | **3.7** | **35.8** | **0.5** | **19.2** | **1.75** | **12** |

## 5.2 Analyzing Search Methods

Fig. 5.a shows the variation of the energy function (Eq. 1) during the search process for 3DLaneNAS, random search, and a local search method [34]. Note that random search is selected as the comparison baseline since it can find the optimal architecture in several applications [19]. 3DLaneNAS provides a continuous reduction in energy function during the search procedure, indicating the proposed NAS method's potential for learning the best architecture. On the other hand, random search and local search methods could not find many improved architectures, which means that our proposed search space is not the only reason behind the efficiency of 3DLaneNAS. Fig. 5.b shows the error-latency trade-off for the best-discovered architectures proposed by different lane detection methods. Results show that 3DLaneNAS provides a higher error–latency trade-off compared to state-of-the-art lane detection methods.

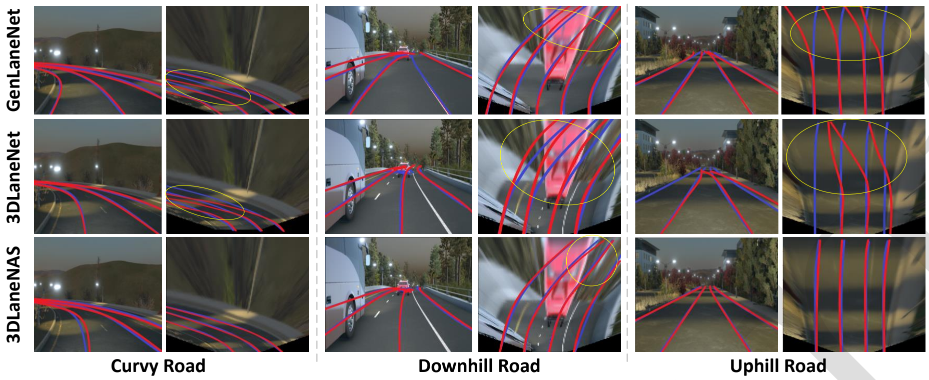


Fig. 4: Illustrating the performance of 3DLaneNet [10], GenLaneNet [11] and 3DLaneNAS in three different road scenarios. Blue lines are ground-truth, red lines are network predictions. Left column: curvy roads' results. Middle column: downhill sample results. Right column: uphill road sample. Yellow circles show low-confidence estimations.

### 5.3 Statement of Reproducibility

A common issue in many NAS studies is to demonstrate reproducibility [19]. To prove the reproducibility of the results, we rerun the 3DLaneNAS five more times with different random seeds. Then, we plot the average energy function for the improved solutions for five times running with the shades to denote the confidence intervals (Fig. 6). According to the Results, while the confidence interval is wide in some iterations, all search runs converge to a similar energy value. The standard deviation (STDEV) is 2.74%. Finally, 3DLaneNAS is an open-source project. The code will be made public upon acceptance.

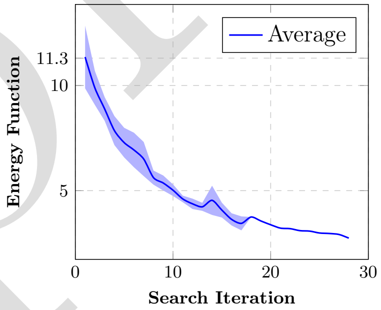
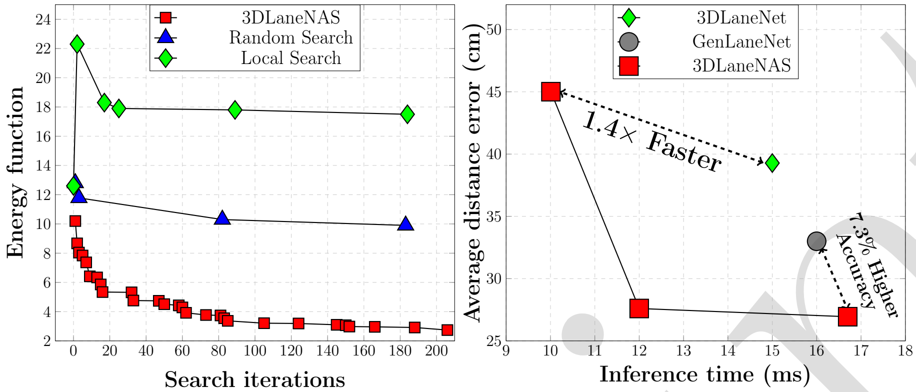


Fig. 6: Demonstrating the reproducibility of results. The solid line shows the average energy value of five runs with different random seeds. The shade is a representation of the STDEV.

## 6 Conclusion

This paper proposes 3DLaneNAS, a multi-objective NAS method for designing a fast and accurate monocular 3D lane detection architecture. 3DLaneNAS improves the performance of monocular 3D lane detection by employing multi-objective simulated annealing as the search method to optimize feature extraction and feature fusion modules. According to experimental results, 3DLaneNAS yields a minimum of 5.2% higher Average Precision and ≈1.33× lower inference



(a) Search Convergency (b) Search Efficiency

Fig. 5: (a). Convergency of energy function (b). Error-Latency trade-off of 3DLaneNAS in comparison with 3DLaneNet and GenLaneNet.

time over counterparts. These results suggest that 3DLaneNAS is an effective method that paves the way for designing efficient lane detection methods.

## ACKNOWLEDGEMENT

This work was supported by Sweden's Innovation Agency (VINNOVA) within the AutoDeep project, and KKS within the DPAC project. This work has been also partially conducted in the project ICT programme which was supported by the European Union through the European Social Fund.

## References

1. Amine, K.: Multiobjective simulated annealing: Principles and algorithm variants. Advances in Operations Research **2019** (2019)
2. Bai, M., Mattyus, G., Homayounfar, N., Wang, S., Lakshmikanth, S.K., Urtasun, R.: Deep multi-sensor lane detection. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 3102–3109. IEEE (2018)
3. Borji, A.: Vanishing point detection with convolutional neural networks. arXiv preprint arXiv:1609.00967 (2016)
4. Cai, H., Gan, C., Wang, T., Zhang, Z., Han, S.: Once-for-all: Train one network and specialize it for efficient deployment. arXiv preprint arXiv:1908.09791 (2019)
5. Chen, P.Y., Lee, C.M., Yeh, H.Z., Huang, Y.C.: Design and implementation for a vision-guided wheeled mobile robot system. In: 2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW). pp. 1–5 (2018). `https://doi.org/10.1109/ICCE-China.2018.8448543`
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009). `https://doi.org/10.1109/CVPR.2009.5206848`

7. Dong, X., Liu, L., Musial, K., Gabrys, B.: Nats-bench: Benchmarking nas algorithms for architecture topology and size. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2021). https://doi.org/10.1109/TPAMI.2021.3054824
8. Elsken, T., Metzen, J.H., Hutter, F.: Neural architecture search: A survey. The Journal of Machine Learning Research **20**(1), 1997–2017 (2019)
9. Foundation, B.: Home of the blender project - free and open 3d creation software, https://www.blender.org/
10. Garnett, N., Cohen, R., Pe'er, T., Lahav, R., Levi, D.: 3d-lanenet: end-to-end 3d multiple lane detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2921–2930 (2019)
11. Guo, Y., Chen, G., Zhao, P., Zhang, W., Miao, J., Wang, J., Choe, T.E.: Genlanenet: A generalized and scalable approach for 3d lane detection. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16. pp. 666–681. Springer (2020)
12. Gurghian, A., Koduri, T., Bailur, S.V., Carey, K.J., Murali, V.N.: Deeplanes: End-to-end lane position estimation using deep neural networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 38–45 (2016). https://doi.org/10.1109/CVPRW.2016.12
13. He, X., Zhao, K., Chu, X.: Automl: A survey of the state-of-the-art. Knowledge-Based Systems **212**, 106622 (2021)
14. Hsu, C.H., Chang, S.H., Liang, J.H., Chou, H.P., Liu, C.H., Chang, S.C., Pan, J.Y., Chen, Y.T., Wei, W., Juan, D.C.: Monas: Multi-objective neural architecture search using reinforcement learning. arXiv preprint arXiv:1806.10332 (2018)
15. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. Advances in neural information processing systems **28** (2015)
16. Jung, J., Bae, S.H.: Real-time road lane detection in urban areas using lidar data. Electronics **7**(11), 276 (2018)
17. Kheyrollahi, A., Breckon, T.P.: Automatic real-time road marking recognition using a feature driven approach. Machine Vision and Applications **23**(1), 123–133 (2012)
18. Lee, S., Kim, J., Shin Yoon, J., Shin, S., Bailo, O., Kim, N., Lee, T.H., Seok Hong, H., Han, S.H., So Kweon, I.: Vpgnet: Vanishing point guided network for lane and road marking detection and recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 1947–1955 (2017)
19. Lindauer, M., Hutter, F.: Best practices for scientific research on neural architecture search. Journal of Machine Learning Research **21**(243), 1–18 (2020)
20. Liu, C., Chen, L.C., Schroff, F., Adam, H., Hua, W., Yuille, A.L., Fei-Fei, L.: Autodeeplab: Hierarchical neural architecture search for semantic image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 82–92 (2019)
21. Liu, H., Simonyan, K., Yang, Y.: Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055 (2018)
22. Loni, M., Mousavi, H., Riazati, M., Daneshtalab, M., Sjödin, M.: Tas:ternarized neural architecture search for resource-constrained edge devices. In: Design, Automation & Test in Europe Conference & Exhibition DATE'22, 14 March 2022, Antwerp, Belgium. IEEE (March 2022), http://www.es.mdh.se/publications/6351-
23. Loni, M., Sinaei, S., Zoljodi, A., Daneshtalab, M., Sjödin, M.: Deepmaker: A multi-objective optimization framework for deep neural networks in embedded systems. Microprocessors and Microsystems **73**, 102989 (2020)

24. Loni, M., Zoljodi, A., Maier, D., Majd, A., Daneshtalab, M., Sjödin, M., Juurlink, B., Akbari, R.: Densedisp: Resource-aware disparity map estimation by compressing siamese neural architecture. In: 2020 IEEE Congress on Evolutionary Computation (CEC). pp. 1–8 (2020). https://doi.org/10.1109/CEC48606.2020.9185611
25. Loni, M., Zoljodi, A., Majd, A., Ahn, B.H., Daneshtalab, M., Sjödin, M., Esmaeilzadeh, H.: Faststereonet: A fast neural architecture search for improving the inference of disparity estimation on resource-limited platforms. IEEE Transactions on Systems, Man, and Cybernetics: Systems pp. 1–13 (2021). https://doi.org/10.1109/TSMC.2021.3123136
26. Loni, M., Zoljodi, A., Sinaei, S., Daneshtalab, M., Sjödin, M.: Neuropower: Designing energy efficient convolutional neural network architecture for embedded systems. In: International Conference on Artificial Neural Networks. pp. 208–222. Springer (2019)
27. Mallot, H.A., Bülthoff, H.H., Little, J., Bohrer, S.: Inverse perspective mapping simplifies optical flow computation and obstacle detection. Biological cybernetics **64**(3), 177–185 (1991)
28. Nedevschi, S., Oniga, F., Danescu, R., Graf, T., Schmidt, R.: Increased accuracy stereo approach for 3d lane detection. In: 2006 IEEE Intelligent Vehicles Symposium. pp. 42–49. IEEE (2006)
29. Nedevschi, S., Schmidt, R., Graf, T., Danescu, R., Frentiu, D., Marita, T., Oniga, F., Pocol, C.: 3d lane detection system based on stereovision. In: Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749). pp. 161–166. IEEE (2004)
30. Padilla, R., Netto, S., da Silva, E.: A survey on performance metrics for object-detection algorithms (07 2020). https://doi.org/10.1109/IWSSIP48289.2020
31. Pizzati, F., García, F.: Enhanced free space detection in multiple lanes based on single cnn with scene identification. In: 2019 IEEE Intelligent Vehicles Symposium (IV). pp. 2536–2541 (2019). https://doi.org/10.1109/IVS.2019.8814181
32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
33. Tang, J., Li, S., Liu, P.: A review of lane detection methods based on deep learning. Pattern Recognition **111**, 107623 (2021)
34. White, C., Nolen, S., Savani, Y.: Exploring the loss landscape in neural architecture search. In: Uncertainty in Artificial Intelligence. pp. 654–664. PMLR (2021)
35. Xu, H., Wang, S., Cai, X., Zhang, W., Liang, X., Li, Z.: Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16. pp. 689–704. Springer (2020)
36. Zhang, W., Mahale, T.: End to end video segmentation for driving: Lane detection for autonomous car. arXiv preprint arXiv:1812.05914 (2018)
37. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578 (2016)
38. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8697–8710 (2018)