Pattern Recognition Letters 185 (2024) 175–183

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Contrastive Learning for Lane Detection via cross-similarity

Ali Zoljodi [a,*], Sadegh Abadijou [b], Mina Alibeigi [c], Masoud Daneshtalab [a,d]

[a] Malardalen University, Universitetsplan 1, Vasteras, 722 20, Sweden
[b] University of Leicester, University Rd, Leicester LE1 7RH, Storbritannien, Leicester, UK
[c] Zenseact AB, Lindholmspiren 2, Goteborg, 417 56, Sweden
[d] Tallin University of Technology, Taliin, Estonia

ARTICLE INFO

Editor: Wu Lin Yuanbo

MSC:
41A05
41A10
65D05
65D17

Keywords:
Contrastive learning
Lane detection
Convolutional neural networks

ABSTRACT

Detecting lane markings in road scenes poses a significant challenge due to their intricate nature, which is susceptible to unfavorable conditions. While lane markings have strong shape priors, their visibility is easily compromised by varying lighting conditions, adverse weather, occlusions by other vehicles or pedestrians, road plane changes, and fading of colors over time. The detection process is further complicated by the presence of several lane shapes and natural variations, necessitating large amounts of high-quality and diverse data to train a robust lane detection model capable of handling various real-world scenarios.

In this paper, we present a novel self-supervised learning method termed Contrastive Learning for Lane Detection via Cross-Similarity (CLLD) to enhance the resilience and effectiveness of lane detection models in real-world scenarios, particularly when the visibility of lane markings are compromised. CLLD introduces a novel contrastive learning (CL) method that assesses the similarity of local features within the global context of the input image. It uses the surrounding information to predict lane markings. This is achieved by integrating local feature contrastive learning with our newly proposed operation, dubbed *cross-similarity*.

The local feature CL concentrates on extracting features from small patches, a necessity for accurately localizing lane segments. Meanwhile, cross-similarity captures global features, enabling the detection of obscured lane segments based on their surroundings. We enhance cross-similarity by randomly masking portions of input images in the process of augmentation. Extensive experiments on TuSimple and CuLane benchmark datasets demonstrate that CLLD consistently outperforms state-of-the-art contrastive learning methods, particularly in visibility-impairing conditions like shadows, while it also delivers comparable results under normal conditions. When compared to supervised learning, CLLD still excels in challenging scenarios such as shadows and crowded scenes, which are common in real-world driving.

1. Introduction

Lane detection is a crucial task in computer vision, particularly for autonomous vehicles and advanced driver assistance systems. This process becomes even more challenging in diverse real-world scenarios, primarily because lane markings are inherently long, thin structures characterized by strong shape priors but limited appearance clues [1]. The visibility of these markings is frequently compromised by adverse factors such as poor lighting conditions, occlusions, and the fading of their color, all of which contribute to making lane detection a highly demanding task [2]. Feature extraction is a crucial component of computer vision algorithms [3,4], especially for lane detection. However, it is not only important to extract features but also to capture the long-range dependencies between these features. This is essential for predicting lanes in segments where visibility is low. Many novel lane detection approaches adopt pixel-level image segmentation techniques [1,5] to enhance the precision of lane detection. In these methods, pixels are labeled either as part of the lane or background.

To develop a robust lane detection method capable of handling natural variations, a significant amount of training data is required. Large-scale labeling of lane markings in road scenes is costly and requires a lot of human labor. However, an abundance of unlabeled data is available, which can simply be used to boost the performance of the model. Furthermore, the appearance of lanes varies across the globe. Unsupervised and self-supervised *contrastive learning* (CL) methods have been proposed for training *deep neural networks* (DNNs) with minimal labeled data and a vast amount of unlabeled data [7,8].

CL methods can be divided into two categories based on the type of representations or features they focus on in an image: local feature

* Corresponding author.
E-mail addresses: ali.zoljodi@mdu.se (A. Zoljodi), ma1023@leicester.ac.uk (S. Abadijou), mina.alibeigi@zenseact.com (M. Alibeigi), masoud.daneshtalab@mdu.se (M. Daneshtalab).

https://doi.org/10.1016/j.patrec.2024.08.007
Received 16 November 2023; Received in revised form 5 June 2024; Accepted 13 August 2024
Available online 20 August 2024

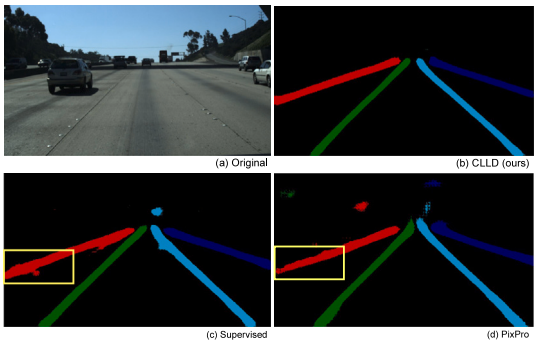

Fig. 1. Comparison of the state-of-the-art segmentation-based lane detection RESA [5] with three different pretraining strategies. (a) Input image (b) RESA output with CLLD (ours) and (c) RESA output with supervised and (d) RESA output with PixPro [6] pretraining. Yellow boxes represent accuracy drops in the detection of lanes that are occluded by cars.

methods and global feature methods. Global feature CL methods are not considered the most effective strategy for lane detection, considering they compare the entirety of an input image with other images. However, for lane detection, it is crucial to localize specific lane segments within the same input image by contrasting different parts of it. On the other hand, local feature CL methods are designed to learn features that classify smaller portions of the input image. These methods can be effectively employed in object localization tasks like lane detection since they focus on specific areas within the input image, enabling more precise detection and localization of objects like lane lines [8].

Existing local CLs are not adequate for lane detection. They are designed to detect completely visible objects and do not have any mechanism to predict the existence of objects that are obscured due to natural variations or occluding by vehicles and pedestrians. To effectively detect and pinpoint lane markings, even in low visibility areas, we propose *contrastive learning for lane detection via cross-similarity* (CLLD), a self-supervised learning method for lane detection. CLLD is a multi-task contrastive learning approach. It trains *convolutional neural networks* (CNNs) to segment an input image and predict masked parts of an image using their surrounding parts. To train the model, we provide both an input image and its augmented version to an encoder. We then measure the consistency between the original image's feature map and the feature map generated from the augmented version that warped to its original shape.

The utilization of masking to self-supervised learning models is referenced in notable works [9], which propose the masking models for training robust image classification models. However, a noteworthy challenge arises when the masked area is sufficiently large, leading to the CNN's inability to perform effectively, as CNNs inherently possess strong inductive bias. In addressing this issue, we introduce a novel operation named cross-similarity, a lightweight operation designed to leverage the similarities between the feature maps surrounding the masked area and their corresponding feature map in the original image. This innovative approach mitigates the loss of important features, thereby enhancing the CNN's capability to effectively detect and reconstruct objects even in scenarios characterized by significant occlusion. We compute the cross-similarity of every patch of the feature map in the masked image with all parts of the feature map in the original image and contrast it with the cross-similarity of each patch in the original image feature maps, and the entire masked image features maps. Differing from [10,11], which mask data to reduce size and focus attention, CLLD uses masking to augment data and train a model to predict missing parts, to improve resiliency against occlusion and data loss.

We assess CLLD on U-Net [12], a popular encoder–decoder that is widely utilized for lane detection tasks. In addition to U-Net, we evaluate the proposed method on RESA [5] and CLRNet [13] as SOTA segmentation-based and anchor-based lane detection methods, respectively. CLLD yields an average 1% improvement in all evaluation metrics over state-of-the-art CL methods on two of the most well-known lane detection datasets, CuLane [1] and TuSimple [14]. To demonstrate its efficacy, we have tested CLLD on the shadow subset of CuLane, which is known to be a challenging set due to its varying light conditions. Our findings show an impressive over 4% improvement in detecting lanes in shadow situations. From the qualitative results shown in Fig. 1, we can see that CLLD outperforms SOTA local CL and supervised learning. Specifically, CLLD is more effective at dealing with occluded parts of a lane.

The main contributions of this work are as follows: **(I)** We demonstrate that previous self-supervised learning approaches may not be the most effective approach for the lane detection task. We highlight some reasons that may contribute to this performance reduction. **(II)** We present the cross-similarity approach, a lightweight operation that computes the correlation between spatial parts of a picture that may contain lane markings and connects them together. **(III)** We propose a novel approach to self-supervised learning for lane detection. Our approach leverages cross-similarity to pretrain lane detection to better detect occluded or worn-out lane segments. **(IV)** We show that our method surpasses supervised learning in detecting lanes under challenging conditions, like shadow-covered markings, by comparing CLLD's performance with that of supervised learning. **(V)** We demonstrate how our method can outperform supervised learning for lane detection in challenging scenarios, such as lane markings concealed by shadows, by comparing CLLD performance with supervised learning.

2. Related work

2.1. Lane detection

Lane detection is a critical module for autonomous driving. The safety of autonomous vehicles is greatly affected by the accuracy and latency of lane detection methods. Lane detection methods are classified as conventional [15] or based on CNN [1,5,13]. Conventional lane detection [15] relies on manual features to identify lanes, limiting their accuracy in different road scenarios. To detect lane segments, Kang and Jung [16] combines local line extraction with dynamic programming to enhance the performance of Hough Transforms, which often struggle in complex scenes. Babu et al. [17] utilizes advanced feature extraction methods, such as LGBPHS and MTP, and optimizes the classification process with a BI-GRU, enhanced by Self-Improved Honey Badger Optimization, to accurately identify lane lines under various conditions, a technique that parallels our focus on robust lane detection.

CNN technology has enabled new solutions for lane detection, such as U-Net [18]. However, challenges arise when detecting occluded lanes due to biases and spatial information capture limitations. Spatial CNN [1] and Recurrent Feature-Shift Aggregator [5] address these challenges by using message-passing to propagate spatial information. Anchor-based [13] lane detection options are also some lane detection methods that behave lanes as a chain of anchors. 3D lane detection offers an alternative approach to solving lane detection challenges by using road curvature patterns and aligning lane markings to these patterns. Janakiraman et al. [19] propose a novel 3D lane detection method that utilizes improved feature extraction techniques and optimized BI-GRU classifiers. This paper discusses how CLLD can enhance the accuracy and robustness of segmentation-based and anchor-based lane detection.

2.2. Contrastive learning

2.2.1. Global features

Global feature methods can aid in image classification by comparing positive and negative samples. Studies such as [20], SimCLR [7], and MoCo [21] use different techniques to train the network to produce similar representations for all views of a sample. However, these methods may not work well for identifying specific parts of an image, as noted by Xie et al. [6]. To improve the quality of self-supervised learned feature representation, Li and Ralescu [22] integrates Bregman divergence into contrastive learning to enhance the learning of distance features between the latent features in the embedding space.

2.2.2. Local features

Local features are proposed to overcome pixel-level and region-level classifications. Different levels of local feature methods can be utilized, such as at the feature-level [8,23], pixel-level [6,24], or region-level [25]. DenseCL [23] is a method that discriminates at the feature-level, inspired by MoCo-V2 [21]. PixPro [6] is a method that discriminates at the pixel-level, inspired by BYOL [26]. The positive samples are identified by pixels with Euclidean distance smaller than a threshold. VICRegL [8] is a trade-off of global and local features that aims to achieve a balance in representation learning. Detecting obscured lanes can be difficult with local feature CLs as they only extract visible segments. However, our method, CLLD, is capable of not only extracting visible parts of lanes but also predicting the existence of obscured lanes based on their surrounding visible parts.

3. Methodology

CLLD is a self-supervised approach that enhances lane detection. It focuses on understanding relationships between different image patches, enabling the detection of occluded or low-visibility lane segments by analyzing their surroundings. This method effectively improves lane detection accuracy when lane markings are poorly visible. Utilizing the concept of CLLD, lane detection encoders are pretrained to reconstruct less visible lane segments by solving Eq. (1).

$$\mathcal{W}^* = \arg\min_{\mathcal{W}} \sum_{p \in P} \mathcal{L}_{clld}(\mathcal{F}(\mathcal{I}_p), \mathcal{F}(\mathcal{M}(\mathcal{I}_p))) \tag{1}$$

The variable $\mathcal{W}^*$ denotes the optimized weights for the lane detection encoder, and $\mathcal{I}$ is the training input. The optimization problem involves extracting local features by dividing the input $\mathcal{I}$ into P patches. The contrastive loss $\mathcal{L}(.)$ is applied to each patch $p \in P$, which undergoes two passes through the CNN backbone $\mathcal{F}$. One pass is in the original shape $\mathcal{F}(\mathcal{I}_p)$, and the other pass is in the masked shape $\mathcal{F}(\mathcal{M}(\mathcal{I}_p))$. The objective of the optimization problem is to minimize the summation of loss values for all patches, denoted as $\sum_{p \in P}(.)$.

We consider lanes as objects with strong shape priors, adhering to a consistent pattern, yet occasionally exhibiting invisibility in random sections. Empirical studies have shown that masking specific areas of the input image and training the encoder to predict those parts can be a beneficial method for teaching the lane detection backbone to predict occluded or missing lane segments. Consequently, we adopt this technique by masking the input image to generate a second view, which is then employed by the contrastive learning method. Below, we provide a comprehensive explanation of the masking that we employed.

In local feature CL, after extracting features, the method wraps back augmentations to position the features in their original location on the image. This technique enables a direct comparison between the extracted local features and their original counterparts. Such a comparison is essential for learning the variations among features from different segments of the same image, thereby enhancing the method's efficacy in feature analysis and interpretation. When using masking, it is not possible to warp areas that have been masked, as no features are extracted from those sections. Our proposal method involves a crucial step to overcome the aforementioned difficulty — enhancing the extracted feature maps with `cross-similarity` information. This module calculates the similarity between each patch of the feature map from the original image, denoted as $\mathcal{F}(\mathcal{I}_P)$, and all patches from the feature map of the masked image, denoted as $\{\forall p \in P : \mathcal{F}(\mathcal{M}(\mathcal{I}_p))\}$. It also compares each patch of the feature map from the masked image, $\{\mathcal{F}(\mathcal{M}(\mathcal{I}_p))\}$, to all patches from the feature map of the original image, denoted as $\{\forall p \in P : \mathcal{F}(\mathcal{I}_p)\}$.

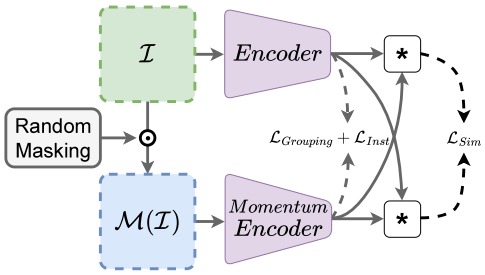

Fig. 2. The CLLD framework.

Through `cross-similarity`, each patch $\mathcal{F}(\mathcal{I}_P)$ has the ability to interact with all the patches in the cross-view, thereby maintaining the positional information of the patches. `cross-similarity` allows for the use of the local feature CL on masked images. This is done by sharing information from each patch with all the other patches in the cross-view, particularly the corresponding patch. In Section 3.3, we provide a comprehensive explanation of the `cross-similarity`.

3.1. Contrastive Learning for Lane Detection

Our method (CLLD) employs momentum contrastive learning [6] (Fig. 2). Given the input image $\mathcal{I}$, the masking function masks some patches with the size $\rho \times \rho$ producing a masked view of the input, denoted as $\mathcal{M}(\mathcal{I})$.

We input $\mathcal{I}$ and $\mathcal{M}(\mathcal{I})$ into two different encoders. The first encoder updates its weights using gradient descent, while the second one updates its weights using the momentum of the first encoder. The output of $\mathcal{M}(\mathcal{I})$ is a feature map $\mathcal{F}(\mathcal{M})(\mathcal{I})$ that may not contain valid information for the masked areas. To enhance the masked areas with comparable information, we compute the `cross-similarity` of each patch from $\mathcal{F}(\mathcal{M}(\mathcal{I}))$ and the entire feature map of the input $\mathcal{F}(\mathcal{I})$. We perform the inverse operation on each patch within $\mathcal{F}(\mathcal{I})$ and the entirety of $\mathcal{F}(\mathcal{M}(\mathcal{I}))$, as discussed above.

3.2. Masking

In this study, similar to masked image modeling approach [9], we mask random portions of inputs and train the model to predict masked parts. This approach trains lane detection backbones to predict hidden objects based on their surroundings an essential application for lane detection algorithms when dealing with occluded or vanished lane markings. The random locations are square patches (see Fig. 3) (with size $\rho \times \rho$) of input images. For a given input $\mathcal{I} \in [0,1]^{H \times W \times C}$ and portions to mask $\mathcal{M} = \{[0,1]^{i \times j}, where\ i \subset \{0, H/P\}\ and\ j \subset \{0, W/P\}\}$, the masked image is generated using Eq. (2).

$$\mathcal{M}(\mathcal{I}) = \begin{cases} \mathcal{N}(0,1) & \text{if } \mathcal{M} \\ \mathcal{I} & otherwise \end{cases} \tag{2}$$

where $\mathcal{M}(\mathcal{I})$ is the masking image. To replace the pixel in a patch, we select a value from the normal distribution $\mathcal{N}(0,1)$.

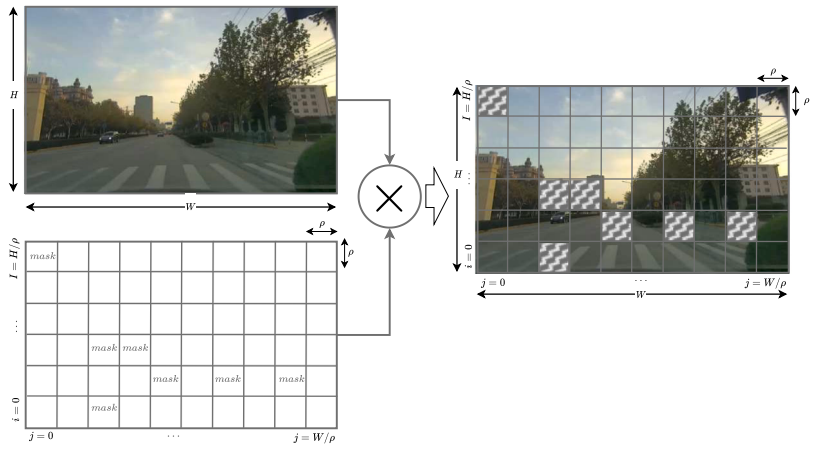

Fig. 3. Masking input image; The given input with size $H \times W$ is divided into $\rho \times \rho$ patches. Each pixel of the masked patch got a random value from a zero-mean normal distribution $\mathcal{N}(0, 1)$.

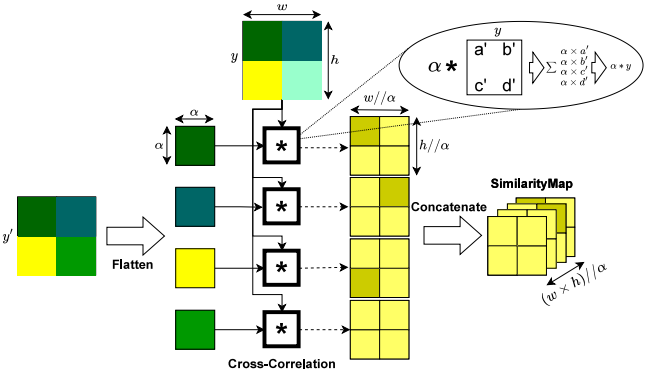

Fig. 4. The cross-similarity operation.

3.3. Cross-similarity

To uncover the local features of masked patches, we utilize the similarity between their surrounding features and their corresponding features in the original image by applying `cross-similarity` operation (Fig. 4).

Assume that the output obtained from feeding the original input image I is y and the output obtained from feeding masked image $\mathcal{M}(I)$ is y'. In order to extract local features, the contrastive loss needs to be applied to the small patches of y and y'. These patches are denoted by $\{y_p \ and \ y'_p, \forall p \in P\}$, where P represents all patches within a feature map. The `cross-similarity` between the full feature map of the first view and a patch of the feature map of the second view y'_p is calculated through the use of Eq. (3).

$$CS(y, y'_p)[i,j] = \sum_{u=-\frac{\alpha}{2}}^{\frac{\alpha}{2}} \sum_{v=-\frac{\alpha}{2}}^{\frac{\alpha}{2}} y'_p[u,v]y[i+u, j+v] \quad \forall i \in [0, h//\alpha] \ \ and \ \ \forall j \in [0, w//\alpha] \tag{3}$$

The variable α represents the measurement of the sides of each patch. Additionally, h and w denote the height and width of feature maps, respectively. To generate the complete `cross-similarity` between y and y', we compute the `cross-similarity` between y and every patch of y' ($\forall p \in y'$) and then concatenate them all together (Eq. (4)).

$$CS(y, (\forall p \in y')) = [CS(p_0, y), CS(p_1, y), \dots, CS(p_z, y)] \quad where \ \ z = (w \times h)//\alpha \tag{4}$$

In order to calculate the `cross-similarity` of y with y', denoted as $CS(y', (\forall p \in y))$, we perform the same operation as before but with y and y' swapped.

As shown in Fig. 4, the `cross-similarity` of each patch y'_p with the feature map y generates a tensor. These tensors are then concatenated in their respective order. By using this mechanism, one can not only discover the similarity between each patch from one feature map and all patches from the other but also retain their location information. Therefore, the contrastive loss can reflect the patterns between the location and the similarity value to the neural network. Furthermore, the contrastive loss between the feature maps of the original view and the masked portions in the second view is higher than that of the other sections. Consequently, as the loss value for the masked sections increases, there is a corresponding increase in attention toward predicting these areas.

3.4. Loss function

Three key tasks are considered to design loss function.

3.4.1. Consistency loss

The first objective of CLLD is to extract visible segments from the input. To train the CNN backbone to extract accurate features, we utilize a $\mathcal{L}_{const}$ (Eq. (5)), which contrasts the consistency of the feature maps produced by CNNs for two views (original and masked).

$$\mathcal{L}_{cons} = -\frac{1}{h \times w} \times \sum_{i=1}^{h} \sum_{j=1}^{w} \frac{y_{ij} . y'_{ij}}{\| y_{ij} \|_2 \times \| y'_{ij} \|_2} \tag{5}$$

To evaluate their consistency, we compute the cosine similarity between each pixel on feature map y and its corresponding pixel on feature map y'. A positive cosine similarity value indicates that the features are consistent, implying a similar orientation in the feature space. Conversely, a negative value signifies inconsistency, suggesting that the features are oriented in opposite directions in the feature space. Therefore, to ensure a positive loss value for inconsistencies, we multiply the cosine similarity value by a negative sign. This approach ensures that a higher loss corresponds to greater inconsistency. Finally, we compute the average of the cosine similarities across all pixels in the feature maps. This average represents the overall consistency between the feature maps, providing a single metric that encapsulates the similarity of the entire feature space.

3.4.2. Similarity loss

Another key objective of CLLD approach is to train CNNs to accurately predict features for masked areas. To achieve this objective, we introduce a similarity loss mechanism $\mathcal{L}_{sim}$. This mechanism is designed to quantify the difference between the predicted features of the masked areas and their actual features, thereby guiding the CNN to make more accurate predictions.

The *similarity loss* (Eq. (6)) computes the consistency between two sets of cross-similarities $CS(y,(\forall p \in y'))$ and $CS(y',(\forall p \in y))$.

$$\mathcal{L}_{sim} = -\frac{CS(y,(\forall p \in y')) \quad . \quad CS(y',(\forall p \in y))}{\| CS(y,(\forall p \in y')) \|_2 \times \| CS(y',(\forall p \in y)) \|_2} \tag{6}$$

$\mathcal{L}_{sim}$ is the cosine similarity between two cross similarities. Differences in results for $CS(y,(\forall p \in y'))$ and $CS(y',(\forall p \in y))$ may arise due to the presence of masked areas on the feature map y'. By leveraging cosine similarity, we can detect variations between different patches (Fig. 4). This detection enables us to train our model with $\mathcal{L}_{Sim}$, focusing on predicting the masked areas by detecting sources of inconsistency.

3.4.3. Classification loss

To learn the categorization representation, we employ an instance-level cosine similarity loss $\mathcal{L}_{inst}$ (Eq. (7)).

$$\mathcal{L}_{inst} = 2 - 2\frac{\hat{y} \quad . \quad \hat{y'}}{\| \hat{y} \|_2 \times \| \hat{y'} \|_2} \tag{7}$$

where the $\hat{y}$ and $\hat{y'}$ are normalized vectors of y and y', respectively. The loss function we used in this study is the summation of all aforementioned loss functions (Eq. (8)).

$$\mathcal{L}_{clld} = \mathcal{L}_{cons} + \mathcal{L}_{sim} + \mathcal{L}_{inst} \tag{8}$$

4. Experimental setup

For a fair comparison, all backbones in our study are pretrained on unlabeled ImageNet-1K [27]. The backbone is ResNet50. We evaluate the performance of CLLD approach on pretraining backbones for three lane detection algorithms: U-Net [12] ($\approx 28M\,parameters$), RESA [5]($\approx 25M\,parameters$), and CLRNet [13] ($\approx 25M\,parameters$). All hyperparameters for the lane detection methods remain unchanged in our study. We employ the LARS optimizer, configured with a cosine learning rate schedule. The initial learning rate is set at 1.0. Additionally, we use a batch size of 1024 and a weight decay parameter set to 1e−5. For the masking process, we set ρ to 14. In the momentum encoder, the momentum value starts at 0.99 and increases to 1. We train ResNet50 on six Nvidia® A100-40 GB GPUs for 100 epochs. To study the effect of α, we train ResNet50 with three different α values: 1, 2, and 3. We mask 30% of the original image to generate the masked version. We fine-tune all lane detection algorithms on two Nvidia® RTX A6000 GPUs.

4.1. Benchmarks

We evaluate CLLD on two lane detection benchmarks: CuLane [1] and TuSimple [14].

4.1.1. CuLane

The CuLane includes 55 h of video data, featuring both highways and urban scenarios. CuLane includes nine validation subsets: *Normal*, *Crowd*, *Night*, *Noline*, *Shadow*, *Arrow*, *Hlight*, and *Curve*. Lane marking predictions are represented by 30-pixel wide lines. A prediction is considered a true positive if it has an Intersection over Union (IoU) greater than 0.5; values lower than 0.5 are classified as false positives. In lane detection, a false positive refers to the incorrect identification of a lane where none exists, while a true negative indicates the correct recognition of the absence of a lane. The absence of a prediction for a lane that is labeled in the ground truth is categorized as a False Negative (FP) while predicting lanes that do not exist is classified as a False Positive (FP). Predictions categorized as FP and FN are considered unsuccessful. The common metrics used for benchmarking [1,5,13] on CuLane are $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$, and $F1-measure = \frac{2\times Precision\times Recall}{Precision+Recall}$.

4.1.2. TuSimple

The dataset comprises 3626 videos, each with a resolution of 1280 × 720 pixels and an approximate length of 20 s.

These videos are captured from a camera mounted on the windshield of a vehicle driving on highways under various weather and lighting conditions. The accuracy metric used for benchmarking on TuSimple is defined as $accuracy = \frac{\Sigma_{clip} C_{clip}}{\Sigma_{clip} S_{clip}}$. Here, the correctly predicted points are denoted by C_{clip}, and the total number of lane points in ground truth is represented by S_{clip}.

5. Results

We specifically chose to evaluate CLLD with U-Net because it is a common encoder–decoder architecture used in various methods that treat lane detection as a semantic segmentation problem [28]. Additionally, we tested our method using RESA [5], which is currently the SOTA semantic segmentation lane detection and not based on the U-Net. Such independent validation is crucial to confirm the accuracy of our model. Finally, we conduct an evaluation of CLLD using CLRNet [13], which is recognized as a leading anchor-based method for lane detection.

5.1. Comparison with prior works

5.1.1. U-Net

The results of the lane detection using U-Net with CLLD pretraining, along with comparisons with other CL methods, are presented in Table 1. More comprehensive results are available in Table 7 in the supplementary material. The results indicate that CLLD with ($\alpha = 3$) outperforms all other methods on the TuSimple benchmark. Furthermore, according to most evaluation metrics, all CLLD versions outperform other methods on CuLane. PixPro is the only method that offers better precision than CLLD; however, its recall is ≈2% lower than the average recall achieved by CLLD. Upon comparing CLLD with PixPro, it is observed that CLLD tends to generate a higher number of FP, whereas PixPro is more prone to producing FN. This comparison indicates that CLLD excels in lane extrapolation compared to PixPro, whereas PixPro demonstrates superior performance in lane interpolation. It has been observed that all CLLD variants outperform VICRegL with a large margin despite being trained for 200 fewer epochs. This suggests that VICRegL, known for its trade-off between global and local features, might not be the most suitable choice for lane detection tasks. We aim to accurately detect lanes not only under nominal scenarios but also under more challenging conditions with low visibility, such as scenarios where lane markings are obscured by shadows. CLLD suppresses all other CL methods, achieving an improvement of 7% in such challenging scenarios (As illustrated in Appendix A. Table 7). CLLD markedly enhances lane detection performance in scenarios where the visibility of extensive lane segments is compromised by lighting conditions, such as shadows. This improvement is attributed to the maintenance of long-range dependencies between local features through the cross-similarity module. Consequently, lanes can be reconstructed by leveraging the similarity among illusion-invariant features, including edge and shape features.

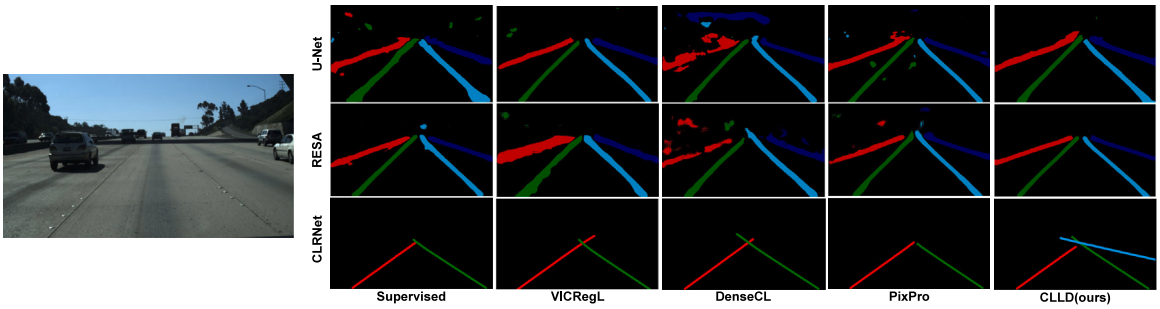

Fig. 5. Qualitative comparison of the results of CLLD with prior SSL methods and supervised learning.

Table 1
Performance of U-Net on CuLane and TuSimple with pretraining by different contrastive learning methods.

Method	# Epoch	CuLane			TuSimple
		Precision	Recall	F1-measure	Accuracy
PixPro [6]	100	73.68	67.15	70.27	95.92
VICRegL [8]	300	67.75	63.43	65.54	93.58
DenseCL [23]	200	63.8	58.4	60.98	96.13
MoCo-V2 [21]	200	63.08	57.74	60.29	96.04
CLLD ($\alpha = 1$)	100	71.98	69.2	70.56	95.9
CLLD ($\alpha = 2$)	100	70.69	69.36	70.02	95.98
CLLD ($\alpha = 3$)	100	71.31	69.59	70.43	96.17

Table 2
Performance of RESA [5] on CuLane and TuSimple with different contrastive learnings.

Method	# Epoch	CuLane			TuSimple
		Precision	Recall	F1-measure	Accuracy
PixPro [6]	100	77.41	73.69	75.51	96.6
VICRegL [8]	300	76.27	69.58	72.77	96.18
DenseCL [23]	200	77.67	73.51	75.53	96.28
MoCo-V2 [21]	200	78.12	73.36	75.66	96.56
CLLD ($\alpha = 1$)	100	79.01	72.99	75.88	96.74
CLLD ($\alpha = 2$)	100	78	73.45	75.66	96.78
CLLD ($\alpha = 3$)	100	78.34	74.29	76.26	96.81

5.1.2. RESA

Table 2 (Appendix Table 8) illustrates the performance of RESA on CuLane and TuSimple with different contrastive learning methods for pretraining. With the RESA architecture, all variations of CLLD surpass the performance of all other methods. CLLD ($\alpha = 1$) emerges as the best precision on the CuLane benchmark. CLLD ($\alpha = 3$) also outperforms other methods on the TuSimple benchmark and, for the most part, on CuLane, according to various evaluation metrics. Similar to U-Net, the combination of RESA and CLLD shows a significant improvement (≈4%) on the shadow subset of CuLane. This highlights the general enhancement in detecting lanes under low visibility conditions. The behavior of CLLD on RESA can be understood through the same rationale applied to U-Net, suggesting a reinterpretation of its efficacy in enhancing lane detection under challenging lighting conditions.

5.1.3. CLRNet

Table 3 (Appendix Table 9) presents the effectiveness of CLLD on CLRNet. Compared to prior contrastive learning methods, CLLD achieves over 1% improvement in recall for CuLane dataset. It also achieves SOTA results on TuSimple accuracy and CuLane's F1-Measure. Similar to previous studies, PixPro achieves better Precision on CuLane. CLRNet is not a semantic segmentation approach. Instead, it detects lane anchors and connects them to achieve better lane extrapolation. CLRNet exhibits marginal improvement with the integration of CLLD. This is attributed to CLRNet's strategy of refining lane detection at a higher layer; if it detects the majority of lane segments, it can extrapolate to fill in missing parts. However, if it fails to identify most of the lane segments, it may disregard the segments it has detected. While CLLD enhances the likelihood of detecting lane segments within CLRNet, the refinement of lanes at a higher level means that detecting discrete lane segments at lower levels may not significantly boost performance.

Table 3
Performance of CLRNet [13] on CuLane and TuSimple with different pretraining strategies.

Method	# Epoch	CuLane			TuSimple
		Precision	Recall	F1-measure	Accuracy
PixPro [6]	100	89.19	70.39	78.67	93.88
VICRegL [8]	300	87.72	71.15	78.72	89.01
DenseCL [23]	200	88.07	69.67	77.8	85.15
MoCo-V2 [21]	200	88.91	71.02	78.96	93.87
CLLD ($\alpha = 1$)	100	88.72	71.33	79.09	90.68
CLLD ($\alpha = 2$)	100	87.95	71.44	78.84	93.48
CLLD ($\alpha = 3$)	100	88.59	71.73	79.27	94.25

5.2. Comparison with supervised learning

Table 4 presents the results of the CLLD pretraining strategy with supervised pretraining. The best improvement (≈1% in the average of all CuLane subsets), compared to supervised learning, is in the RESA with ResNet50 as the backbone. CLLD also achieves a maximum ≈4% increase in CuLane's low visible subsets, such as the shadow. CLLD outperforms supervised learning on RESA for all metrics on both datasets, with the exception of the FP rate on TuSimple. FP is the only metric for which supervised learning has provided better prediction outcomes than CLLD at a rate of 0.0343 per prediction.

CLLD performance is equivalent to supervised learning based on most evaluation metrics on CLRNet (± ≤1%). For the shadow subset of CuLane, the accuracy of CLRNet was about ≈4% better with CLLD pretraining than supervised learning. CLLD performance in U-Net is comparable to supervised learning. It gains over 1% better precision than supervised learning. CLLD also produces over 300 more FPs than supervised learning in the cross subset of CuLane.

5.3. Visual demonstration

Fig. 5 illustrates a qualitative comparison of lane prediction pre-retained on CLLD compared with supervised learning and prior lane detection methods. The results illustrate CLLD performance, especially for the most left lane with an occluder. Most other training strategies detect lanes, but with many false positives, except DenseCL, which destroys the lane. PixPro also has worse predictions than CLLD, with significant FP for the occluded part.

Fig. 6 is the interpolated view of the latent layer on RESA for supervised and self-supervised learning (CLLD). The results show that

Table 4
Comparison of the performance of state-of-the-art lane detection methods on CuLane and TuSimple in two situations of pretraining with supervised learning and CLLD self-supervised learning.

Method	Pretrain	CuLane												TuSimple		
		Overall (%)			F1-measure (%)							FP				
		Precision	Recall	F1	Normal	Crowd	Night	Noline	Shadow	Arrow	Hlight	Curve	Cross	Accuracy	FP	FN
U-Net [12]	Supervised	70.93	69.65	70.28	89.82	67.72	64.95	40.49	68.13	84.48	59.83	67.02	2482	96.24	0.0489	0.0428
	CLLD	71.31	69.59	70.43	89.8	68.39	64.65	40.68	68.86	84.5	58.93	66.2	2656	96.17	0.055	0.045
RESA [5]	Supervised	77.51	73.15	75.27	92.16	73.16	69.99	47.71	72.97	88.16	68.79	70.65	1503	96.67	0.031	0.0265
	CLLD	78.34	74.29	76.26	92.57	74.35	71.21	48.83	76.62	89.14	67.58	72.68	1454	96.81	0.0343	0.0264
CLRNet [13]	Supervised	88.21	71.88	79.22	93.1	77.83	74.3	52.69	76.92	89.63	73.16	69.41	1082	93.17	0.0232	0.0748
	CLLD	88.59	71.73	79.27	92.94	77.44	74.43	53.3	81.2	89.31	72.46	68.4	1026	94.25	0.214	0.069

Table 5
Ablation: Multi-task contrastive learning. Comparison of the impact of similarity loss and consistency loss on the lane detection accuracy.

$\mathcal{L}_{sim}$	$\mathcal{L}_{cons}$	CuLane			TuSimple
		Precision	Recall	F1	Accuracy
*		76.91	70.82	73.74	95.94
	*	77.41	73.69	75.51	95.92
*	*	78.34	74.29	76.26	96.17

Table 6
Ablation: Impact of the masking as the augmentation. Comparison of the accuracy of CLLD with and without using masking as the augmentation.

Masking	CuLane			TuSimple
	Precision	Recall	F1	Accuracy
No	72.2	65.77	68.84	95.7
Yes	71.98	69.2	70.56	96.17

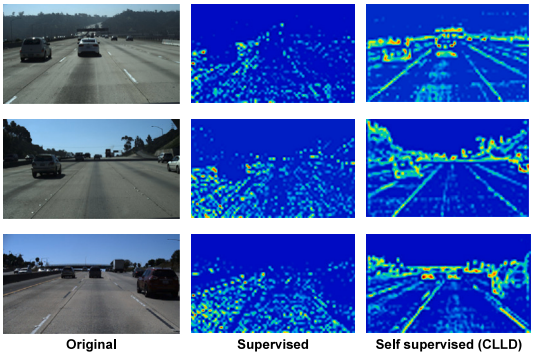

Fig. 6. Low-level features in RESA; The left column is the input image, the middle column is low-level features in supervised learning, and the right column is low-level features in CLLD.

supervised learning pays more attention to the texture of the road; however, CLLD focuses more on lanes' and objects' shapes, which is more important for lane detection. Wu et al. [29] study these differences in supervised and self-supervised learning behavior. This may be a reason why self-supervised learning performs better on lane detection.

6. Ablation study

6.1. Similarity loss impact

Table 5 (Appendix Table 10) ablates CLLD performance with a single similarity loss, a single consistency loss, and the combination of them together. We did not study the absence of instance loss because it did not affect the segmentation results and used it for classification. The accuracy of combining similarity and consistency is remarkably better than using only one. $\mathcal{L}_{sim} + \mathcal{L}_{cons}$ performed significantly better results (≈5%) in challenging subsets of CULane dataset such as the shadow; however, the performance in the normal CuLane subset is not affected by the loss of similarity. This observation illustrates the effect of similarity loss on the detection of low visible lanes.

6.2. Impact of masking as the augmentation

Table 6 (Appendix Table 11) examines the effect of the masking strategy on overall accuracy. CLLD with masking yields a significantly better recall in CuLane (≈4%); however, it achieves a marginally lower precision (≈0.2%). It increases the F-measure by an average of ≈2% over all CuLane subsets. The combination with masking also improves the accuracy of TuSimple markedly (≈1%).

7. Discussion

CLLD framework exhibits comparable performance in scenarios where lane delineations remain unaffected by occlusions or fading phenomena. However, it demonstrates a markedly superior performance in conditions where lane visibility is partially compromised, such as within the shadowed subsections of the CuLane dataset. This enhanced efficacy can be attributed to the incorporation of a cross-similarity module within the encoding process, which facilitates the capture of long-range dependencies across the visual field. Consequently, this mechanism affords the model the capability to infer the presence of lanes even in segments where they have become ostensibly invisible. It achieves this by leveraging the feature comparisons between visible lane segments and their occluded counterparts, thereby providing a probabilistic basis for the accurate prediction of the latter. Such an approach underscores the pivotal role of cross-similarity in enhancing the robustness of lane detection algorithms under varying visibility conditions. Looking forward, we aim to explore the application of CLLD to Vision Transformers (ViTs) and develop a second version of CLLD compatible with both CNNs and ViTs. This expansion will address the evolving challenges in autonomous driving and lane detection technologies, potentially leading to even more robust lane detection systems.

8. Conclusion

Our paper presents a novel self-supervised approach, Contrastive Learning for Lane Detection via cross-similarity (CLLD), designed to enhance the resilience of lane detection models in adverse conditions. CLLD is a multi-task CL that addresses the challenge of detecting obscured lane markings caused by factors like poor lighting and weather by integrating our novel operation cross-similarity to local feature CLs. CLLD captures long-range dependencies between different lane segments through the use of a cross-similarity operation. By computing the similarity between illusion-invariant features such as shape and edges, cross-similarity can reconstruct lane patterns even in segments with low visibility. This approach utilizes similarity to enhance lane detection precision for lanes that are partially occluded or have invisible due to natural variations.

Our method (CLLD) demonstrates remarkable improvements over existing contrastive learning techniques, particularly excelling in scenarios with low visibility, such as shadows. In the future, our focus will be on identifying challenges associated with applying CLLD to Vision Transformers (ViTs) and developing a second version of CLLD that is compatible with both CNNs and ViTs.

CRediT authorship contribution statement

Ali Zoljodi: Writing – review & editing, Writing – original draft. **Sadegh Abadijou:** Writing – review & editing, Writing – original draft. **Mina Alibeigi:** Writing – review & editing, Supervision. **Masoud Daneshtalab:** Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Ali Zoljodi reports financial support was provided by Mälardalen University Sweden. Ali Zoljodi reports a relationship with Mälardalen University Sweden that includes: employment. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This work was supported by Sweden's Innovation Agency (VINNOVA) within the AutoDeep project and the European Union and Estonian Research Council via project TEM-TA138. The computations were enabled by the supercomputing resource Berzelius provided by National Supercomputer Centre at Linkoping University and the Knut and Alice Wallenberg Foundation.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.patrec.2024.08.007.

References

[1] X. Pan, J. Shi, P. Luo, X. Wang, X. Tang, Spatial as deep: Spatial CNN for traffic scene understanding, Proc. AAAI Conf. Artif. Intell. 32 (2018) http://dx.doi.org/10.1609/aaai.v32i1.12301, URL: https://ojs.aaai.org/index.php/AAAI/article/view/12301.

[2] R. Liu, Z. Yuan, T. Liu, Z. Xiong, End-to-end lane shape prediction with transformers, in: 2021 IEEE Winter Conference on Applications of Computer Vision, WACV, 2021, pp. 3693–3701, http://dx.doi.org/10.1109/WACV48630.2021.00374.

[3] Y.B. Özçelik, A. Altan, Overcoming nonlinear dynamics in diabetic retinopathy classification: A robust AI-based model with chaotic swarm intelligence optimization and recurrent long short-term memory, Fract. Fract. 7 (8) (2023) http://dx.doi.org/10.3390/fractalfract7080598, URL: https://www.mdpi.com/2504-3110/7/8/598.

[4] İ. Yağ, A. Altan, Artificial intelligence-based robust hybrid algorithm design and implementation for real-time detection of plant diseases in agricultural environments, Biology 11 (12) (2022) http://dx.doi.org/10.3390/biology11121732, URL: https://www.mdpi.com/2079-7737/11/12/1732.

[5] T. Zheng, H. Fang, Y. Zhang, W. Tang, Z. Yang, H. Liu, D. Cai, RESA: Recurrent feature-shift aggregator for lane detection, Proc. AAAI Conf. Artif. Intell. 35 (2021) 3547–3554, http://dx.doi.org/10.1609/aaai.v35i4.16469, URL: https://ojs.aaai.org/index.php/AAAI/article/view/16469.

[6] Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, H. Hu, Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 16679–16688, http://dx.doi.org/10.1109/CVPR46437.2021.01641.

[7] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: H.D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 1597–1607, URL: https://proceedings.mlr.press/v119/chen20j.html.

[8] A. Bardes, J. Ponce, Y. LeCun, VICRegL: Self-supervised learning of local visual features, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Advances in Neural Information Processing Systems, Vol. 35, Curran Associates, Inc., 2022, pp. 8799–8810, URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/39cee562b91611c16ac0b100f0bc1ea1-Paper-Conference.pdf.

[9] Y. Shi, N. Siddharth, P. Torr, A.R. Kosiorek, Adversarial masking for self-supervised learning, in: K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, S. Sabato (Eds.), Proceedings of the 39th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 162, PMLR, 2022, pp. 20026–20040, URL: https://proceedings.mlr.press/v162/shi22d.html.

[10] L. Shen, H. Tao, Y. Ni, Y. Wang, V. Stojanovic, Improved YOLOv3 model with feature map cropping for multi-scale road object detection, Meas. Sci. Technol. 34 (4) (2023) 045406, http://dx.doi.org/10.1088/1361-6501/acb075.

[11] X. Song, Z. Peng, S. Song, V. Stojanovic, Anti-disturbance state estimation for PDT-switched RDNNs utilizing time-sampling and space-splitting measurements, Commun. Nonlinear Sci. Numer. Simul. 132 (2024) 107945, http://dx.doi.org/10.1016/j.cnsns.2024.107945, URL: https://www.sciencedirect.com/science/article/pii/S100757042400131X.

[12] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W.M. Wells, A.F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham, 2015, pp. 234–241, http://dx.doi.org/10.1007/978-3-319-24574-4_28.

[13] T. Zheng, Y. Huang, Y. Liu, W. Tang, Z. Yang, D. Cai, X. He, Clrnet: Cross layer refinement network for lane detection, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 888–897, http://dx.doi.org/10.1109/CVPR52688.2022.00097.

[14] S. Shirke, R. Udayakumar, Lane datasets for lane detection, in: 2019 International Conference on Communication and Signal Processing, ICCSP, 2019, pp. 0792–0796, http://dx.doi.org/10.1109/ICCSP.2019.8698065.

[15] T.-Y. Sun, S.-J. Tsai, V. Chan, HSI color model based lane-marking detection, in: 2006 IEEE Intelligent Transportation Systems Conference, 2006, pp. 1168–1172, http://dx.doi.org/10.1109/ITSC.2006.1707380.

[16] D.-J. Kang, M.-H. Jung, Road lane segmentation using dynamic programming for active safety vehicles, Pattern Recognit. Lett. 24 (16) (2003) 3177–3185, http://dx.doi.org/10.1016/j.patrec.2003.08.003, URL: https://www.sciencedirect.com/science/article/pii/S0167865503001843.

[17] A. Babu, T. Kavitha, R.P. de Prado, B.D. Parameshachari, M. Woźniak, HOPAV: Hybrid optimization-oriented path planning for non-connected and connected automated vehicles, IET Control Theory Appl. 17 (14) (2023) 1919–1929, http://dx.doi.org/10.1049/cth2.12441, URL: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cth2.12441.

[18] L.-A. Tran, M.-H. Le, Robust U-net-based road lane markings detection for autonomous driving, in: 2019 International Conference on System Science and Engineering, ICSSE, 2019, pp. 62–66, http://dx.doi.org/10.1109/ICSSE.2019.8823532.

[19] B. Janakiraman, S. Shanmugam, R. Pérez de Prado, M. Wozniak, 3D road lane classification with improved texture patterns and optimized deep classifier, Sensors 23 (11) (2023) http://dx.doi.org/10.3390/s23115358, URL: https://www.mdpi.com/1424-8220/23/11/5358.

[20] Y. Tian, D. Krishnan, P. Isola, Contrastive multiview coding, in: A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (Eds.), Computer Vision – ECCV 2020, Springer International Publishing, Cham, 2020, pp. 776–794, http://dx.doi.org/10.1007/978-3-030-58621-8_45.

[21] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 9726–9735, http://dx.doi.org/10.1109/CVPR42600.2020.00975.

[22] Z. Li, A. Ralescu, Generalized self-supervised contrastive learning with bregman divergence for image recognition, Pattern Recognit. Lett. 171 (2023) 155–161, http://dx.doi.org/10.1016/j.patrec.2023.05.020, URL: https://www.sciencedirect.com/science/article/pii/S0167865523001472.

[23] X. Wang, R. Zhang, C. Shen, T. Kong, L. Li, Dense contrastive learning for self-supervised visual pre-training, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 3023–3032, http://dx.doi.org/10.1109/CVPR46437.2021.00304.

[24] X. Zhao, R. Vemulapalli, P.A. Mansfield, B. Gong, B. Green, L. Shapira, Y. Wu, Contrastive learning for label efficient semantic segmentation, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 10603–10613, http://dx.doi.org/10.1109/ICCV48922.2021.01045.

[25] T. Xiao, C.J. Reed, X. Wang, K. Keutzer, T. Darrell, Region similarity representation learning, in: 2021 IEEE/CVF International Conference on Computer Vision, ICCV, 2021, pp. 10519–10528, http://dx.doi.org/10.1109/ICCV48922.2021.01037.

[26] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, M. Valko, Bootstrap your own latent - a new approach to self-supervised learning, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 21271–21284, URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255, http://dx.doi.org/10.1109/CVPR.2009.5206848.

[28] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, Q. Wang, Robust lane detection from continuous driving scenes using deep neural networks, IEEE Trans. Veh. Technol. 69 (1) (2020) 41–54, http://dx.doi.org/10.1109/TVT.2019.2949603.

[29] H. Wu, Y. Gao, Y. Zhang, S. Lin, Y. Xie, X. Sun, K. Li, Self-supervised models are good teaching assistants for vision transformers, in: K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, S. Sabato (Eds.), Proceedings of the 39th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 162, PMLR, 2022, pp. 24031–24042, URL: https://proceedings.mlr.press/v162/wu22c.html.