Lane NET : Enhanced Curved Lane Detection With Deep Matching Process

Team Members –

1) *Advaith Moholkar (B23ES1005)*
2) *Parth Khiriya (B23EE1051)*
3) *Vishrut Aditya Ratnoo (B23EE1102)*
4) *Debjit Ghorai (B23ES1013)*
5) *Shlok Agarwal (B23EE1096)*

Mentor – *Dr. Himanshu Kumar*

Abstract

Lane detection plays a crucial role in autonomous driving by providing vital data to ensure safe navigation. Modern algorithms rely on anchor-based detectors, which are then followed by a label-assignment process to categorize training detections as positive or negative instances based on learned geometric attributes. Accurate label assignment has great impact on the model performance, that is usually relying on a pre-defined classical cost function evaluating GT-prediction alignment. However, classical label assignment methods face limitations due to their reliance on predefined cost functions derived from low-dimensional models, potentially impacting their optimality. Our research introduces LaneNet, a deep learning submodule-based approach aimed at improving the label assignment process. Integrated into a state-of-the-art lane detection network such as the Cross Layer Refinement Network for Lane Detection (CLRNet), LaneNet replaces the conventional label assignment process with a submodule network. The integrated model, CLRLaneNet, surpasses CLRNet, showing substantial improvements in scenarios involving curved lanes.

Introduction

A crucial component of contemporary object identification models, label assignment significantly affects the models' overall performance. The techniques employed to identify. In the context of detection models, assignment can provide wildly different outcomes.

One of the primary bottlenecks in lane detection pipelines is the **label assignment process**—the task of deciding which predicted lanes correspond to which ground truth annotations during training. Existing models typically rely on **fixed, handcrafted cost functions** to perform this task. These cost functions :

1. Operate in low-dimensional space.
2. Inflexible in new scenarios and in new data set.
3. Cannot dynamically adjust based on context.

This rigid structure results in poor detection in **curved and complex scenarios**, as the model cannot correctly learn meaningful lane associations.

Dataset

In our experiments, we conducted research using the **CULane dataset**, a widely used benchmark for lane detection. This dataset contains

- **100,000 images**, divided into three subsets: **training set, validation set, and test set**.
- All **images** are formatted to have dimensions of "**1640x590**" pixels.
- **9 test categories**: Normal, Crowded, Dazzle Light, Shadow, No Line, Arrow, Curve, Crossroad, and Night.

Since our research emphasizes curved lane performance, we also create a **curve-focused sub-dataset** from CULane –

- Contains 8,677 images where only **curved lanes** are visible.
- Used to **pretrain LaneNet** and **fine tune the final model** (CLRLaneNet).

Methodology

In object detection or lane detection, the model needs to learn which predicted objects (lanes, in our case) match with ground truth annotations. This is called **label assignment**.

Traditionally, this is done **using a cost function** that compares geometric properties like:

- Distance between predicted and true start points.
- Orientation (angle) difference.
- Shape alignment or curvature.

This cost assigns a score to each prediction-GT pair, and the top-k scores are chosen as positive samples for training. However, this **method is rigid**, often favouring **high-confidence** but **poorly aligned predictions**.

CLRNet: The Baseline Architecture

We adopted the state-of-the-art CLRNet as the baseline of our model due to its superior performance compared to alternative methods. It works as follows :

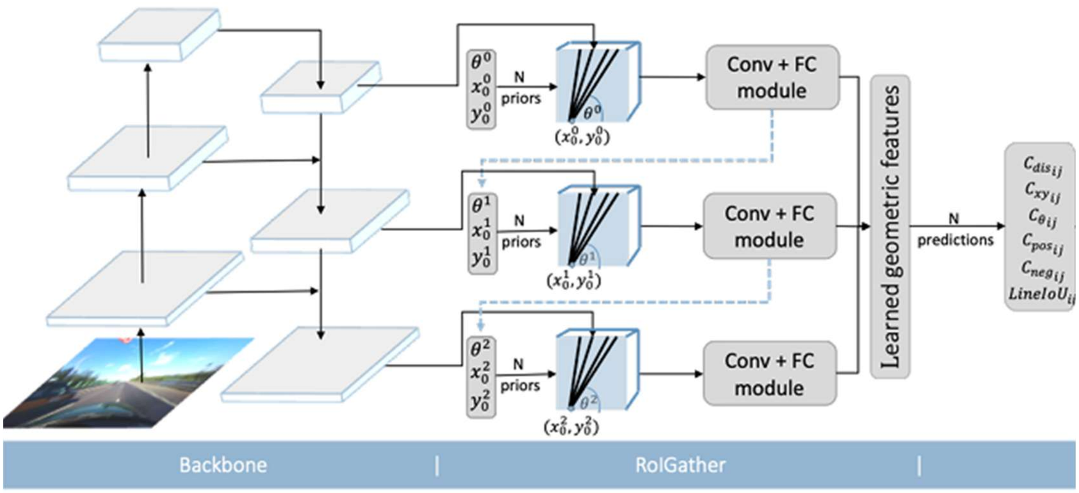
- The model utilizes a **backbone (ResNet or DLA)** to extract features.
- Builds a **Feature Pyramid Network (FPN)** to handle multi-scale lane features.
- Applies **anchor-based regression** where predefined straight-line anchors are refined to match lane shapes.

A lane prediction consists of learned geometric features, including foreground and back ground probabilities, the length of the lane prior, the starting point of the lane prior, the angle between the x-axis of the lane prior, and N offsets, that is, the horizontal distance between the prediction and its lane prior.

Label assignment in CLRNet is done using the **SimOTA** technique from YOLOX, which assigns the top-k predictions per GT based on a composite cost function:

Cost Function:

$$\mathcal{L} = \lambda_0 \mathcal{L}_{xyl\theta} + \lambda_1 \mathcal{L}_{cls} + \lambda_2 \mathcal{L}_{seg} + \lambda_3 \mathcal{L}_{LineIoU}$$



LaneNet : Learning to Match

To address the limitations of handcrafted cost functions, we introduce **LaneNet**, a fully connected **deep neural network** designed to learn the matching function directly.

Inputs to LaneNet :

Each prediction-GT pair is encoded using :

- Distance, angle, and classification score (same as CLRNet).
- LineIoU metric.
- All values are normalized (e.g., angles clamped between 0–180°, x/y between 0–image size).

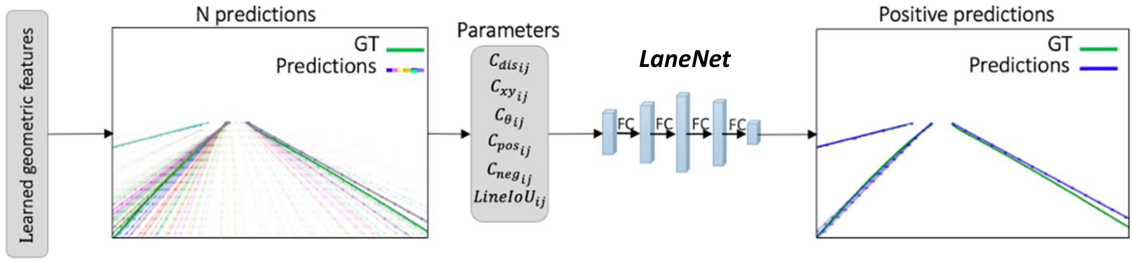
Architecture :

LaneNet follows an **encoder-decoder** style. Encoder-decoder models are used to **handle sequential data, specifically mapping input sequences to output sequences of different lengths**, such as neural machine translation, text summarization, image captioning and speech recognition. In such tasks, mapping a token in the input to one in the output is often indirect.

For example, take machine translation: in some languages, the verb appears near the beginning of the sentence (as in English), in others at the end (such as German) and in some, the location of the verb may be more variable (for example, Latin). An encoder-decoder network generates variable length yet contextually appropriate output sequences to correspond to a given input sequence.

The architecture consists of :

- Two fully connected layers in each stage.
- Leaky ReLU activations
- Final Layer has sigmoid output, denoting match probability



Output

For each prediction-GT pair, LaneNet outputs a score:

- Score ∈ [0,1]
- High scores → better match.

Training LaneNet : The Teacher-Student Approach

LaneNet is trained using a teacher-student framework:

- *Teacher:* Pretrained CLRNet with classical cost function.
- *Student:* LaneNet learns to imitate and improve teacher's decision-making.

A pair is labelled as positive if:

- It has a high match score from CLRNet's cost function.
- Its loss value (LineIoU + classification loss) is below a threshold.

$$match_{ij} = \begin{cases} +1, & \text{if } (C_{ij} = +1) \cap (\mathcal{L}_{ij} < t_L) \\ -1, & \text{else} \end{cases}$$
$$\mathcal{L}_{ij} = \mathcal{L}_{LIoU}(ij) + \lambda \mathcal{L}_{cls}(ij)$$

First, LaneNet was trained using this sub dataset, and subsequently, we fine-tuned CLRNet's results using the pre-trained LaneNet on the same sub dataset. This approach allowed us to leverage the knowledge learned by LaneNet to further refine CLRNet's performance in the curve section.

CLRLaneNet: Final Integrated Architecture

This integrated system **preserves the strengths of CLRNet**—namely its **multi-scale feature detection** and **strong performance on straight lane** segments—while augmenting it with a learnable label assignment module that adapts intelligently to complex road geometries like curves, forks, and intersections.

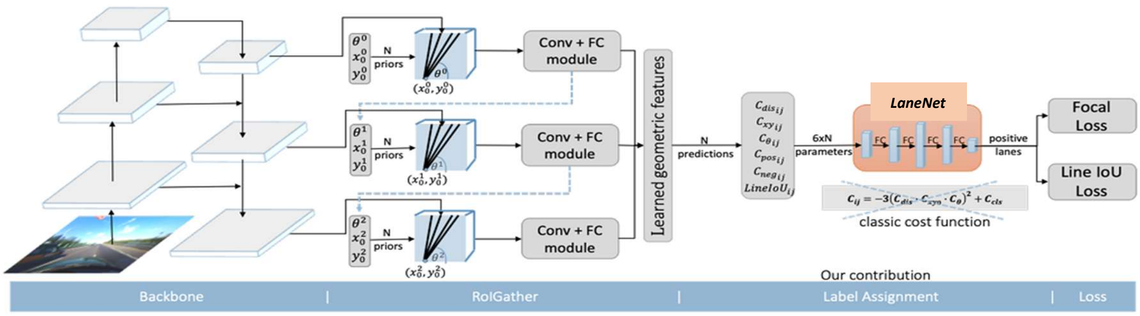
Architecture

CLRLaneNet builds upon the original anchor-based structure of CLRNet. At a high level, here's what happens:

1) *Backbone Feature Extraction*
   - The input image is **processed by a CNN backbone** (e.g., ResNet-34/101 or DLA-34) with a Feature Pyramid Network (FPN) to extract multi-scale features.
2) *Lane Prior Proposal & ROIGather*
   - **Anchor lanes** (called lane priors, defined by x_0, y_0, θ) are **generated** at each scale.
   - These **priors guide** the **ROIGather** module to **pool per-lane features**.
3) *Lane Prediction*
   - Each **lane feature** is passed through a prediction head that outputs :
     - Classification Score
     - Geometric Parameters (offset, length, angle)
     - Segmentation mask (for auxiliary training)
4) *LaneNet – Based Label Assignment*
   - Rather than using a fixed cost function, CLRLaneNet uses **LaneNet**—a small neural network that computes match probabilities between predictions and GT lanes.
   - This score determines which predictions are treated as positive matches during training, improving accuracy in complex lane scenarios like curves.

Final training loss includes :

$$\mathcal{L} = \lambda_0 \mathcal{L}_{xyl\theta} + \lambda_1 \mathcal{L}_{cls} + \lambda_2 \mathcal{L}_{seg} + \lambda_3 \mathcal{L}_{LineIoU}$$



Implementation

Our implementation details are completely identical to the baseline CLRNet. Given below are the conditions which were used when we were training the model :

- The input images were resized to dimensions of 320x800.
- The optimization process incorporates the **AdamW optimizer** and a **cosine learning rate decay**.
- For training, we initialize CLRLaneNet **weights** with the **pre-trained CLR Net's weights** and incorporate the pre-trained LaneNet's weights for an additional 5 epochs to refine the label assignment.
- We adopted a **threshold of 0.7** for **LaneNet's output score** during training, categorizing pairs with scores that surpass this threshold as positive and those that below it as negative.

Evaluation Metrics

The official metric of the CUlane dataset in the evaluation process is the **F1 score**.

During evaluation, first predictions with confidence scores exceeding a predefined threshold are filtered. These selected predictions are then utilized to calculate the final F1 score which is derived from the IoU metric.

In particular, the official metric considers the lanes as 30-pixels-thick lines. If a prediction has an IoU greater than $t_{IoU} = 0.5$ with a GT lane, it is considered a TP, whereas unmatched predictions and GTs are referred to as FP and FN.

The F1 score is then calculated as follows:

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

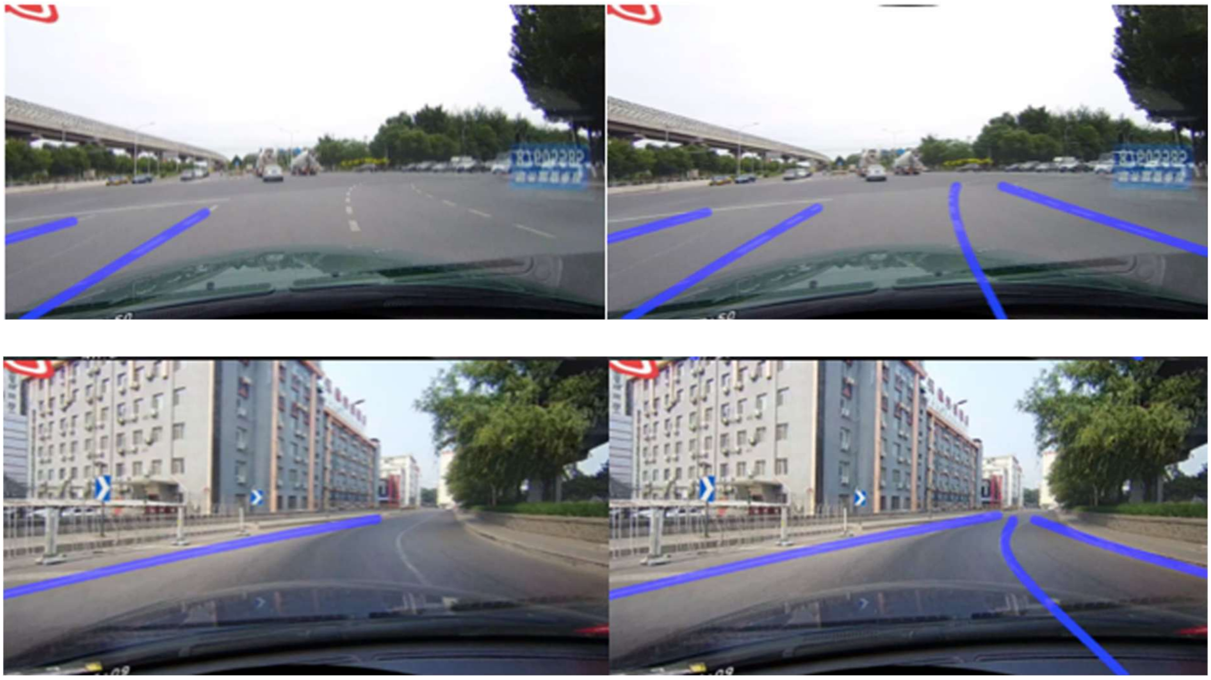
Results

Backbone	***Method***	***Curve F1***	***Improvement***
ResNet34	CLRNet	72.77	-
	CLRLaneNet	***75.57***	***+2.8 %***

ResNet101	CLRNet	75.57	-
	CLRLaneNet	***77.87***	***+2.3 %***
DLA34	CLRNet	74.13	-
	CLRLaneNet	***77.09***	***+2.96 %***

Improvements are most significant in **curve, crowd, and no-line** categories while other sections maintain parity or slight improvements.

Given below are some fundamental examples depicting the difference in CLRNet and CLRLaneNet predictions :



Different lightings and shadows :

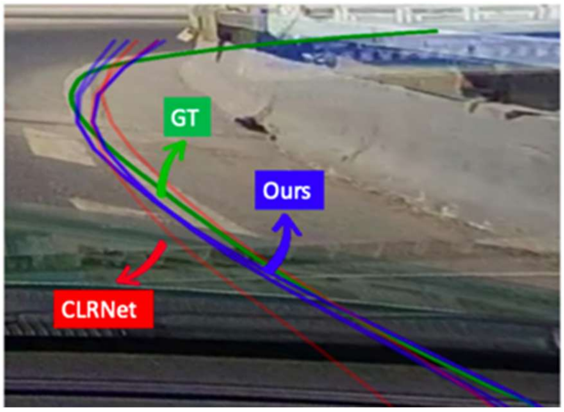
Qualitative Results

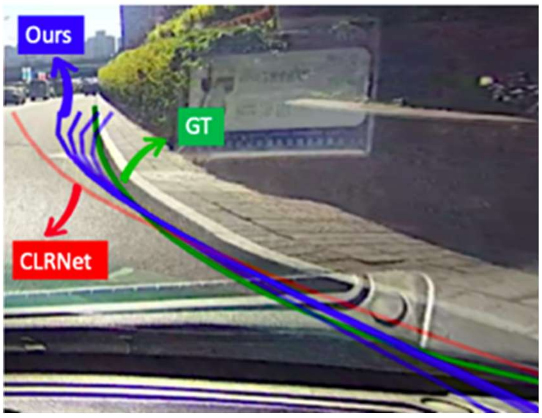
CLRLaneNet consistently finds:

- More geometrically accurate lanes.
- Higher confidence scores.
- Fewer false negatives in difficult cases like night, shadow, and curves.

Confidence Analysis

- Using LaneNet boosts overall detection confidence.
- Enables increasing the evaluation threshold → better precision.





Discussion

False Positives from Incomplete GT

In many frames, lanes visible to humans were not annotated in the GT. LaneNet correctly detects them, but they are counted as false positives—this unfairly lowers the F1 score.

LaneNet Convergence

Occasional loss fluctuations observed—due to automated labelling (not manually cleaned). Despite that, overall training was successful.

Comparison with Other Methods

CLRNet performs well using another classical method (LaneIoU), but it lacks the flexibility and generalizability of a learned model like LaneNet.

Conclusion

In this project, we introduced **LaneNet**, a deep neural network for learning label assignment in lane detection. Integrated with CLRNet to create **CLRLaneNet**, it offers significant gains in curved lane detection and confidence evaluation.

Key Takeaways:

- Classical cost functions are insufficient for complex scenarios.
- LaneNet adapts dynamically and learns from both geometry and confidence.
- Future potential: apply LaneNet-style label assignment to object detection, instance segmentation, or pose estimation.

References

1. *X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang. Spatial as Deep: Spatial CNN for Traffic Scene Understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, 2018*
2. *T. Zheng et al. CLRNet: Cross Layer Refinement Network for Lane Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.*
