Improving Confidence of Lane Detection With LaneIoU

Cs 512 - Computer Vision
Illinois Institute of Technology

Presented by Riddhi Das (A20582829),
Madhur Gusain (A20572395)

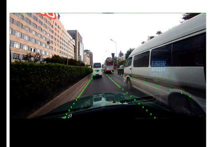
Introduction

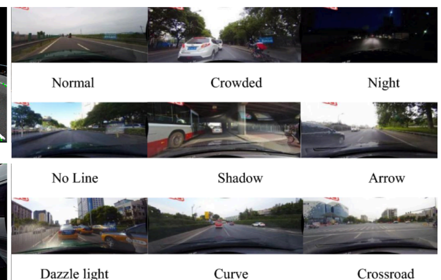
LANE DETECTION IS CRITICAL FOR AUTONOMOUS DRIVING.

The Problem	Traditional detectors predict accurate positions but unreliable confidence scores. Misalignment between confidence and true geometric accuracy.
The Solution	We implement CLRerNet to solve this using LaneIoU.

Key Hurdles Encountered

- Confidence scores don't align with actual IoU.
- Existing similarity metrics (LineIoU) are weak for curves.
- Misleading assignments degrade training.
- Poor generalization on complex lanes.





Project Objectives

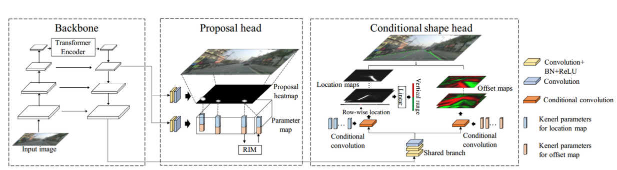
- Implement CLRerNet using MMDetection
- Integrate LaneIoU into training loss, assignment, and confidence supervision.
- Evaluate model performance on CULane dataset.
- Analyze results and improve robustness.

LaneIoU – The Key Idea

- Computes overlap row-wise across image height.
- More accurate than LineIoU for curved and tilted lanes.
- Used for loss computation, sample assignment, and confidence prediction.

Model Architecture

- Backbone: DLA-34
- Neck: Feature Pyramid Network
- Head: Modified proposal head (classification, regression, confidence)
- Uses dynamic-k assignment with LaneIoU



Dataset

- Dataset: CULane
  - 133K frames, 1640×590 resolution
  - Scenarios: Urban, curves, night, occlusion
- Filtered to 62,532 valid samples
- Masks from laneseg_label_w16/

Implementation & Environment

Implementation

- Docker-based training pipeline
- MMDetection v3.0.0 + MMCV v2.0.0+MMEngine v0.7.4

Optimizer

AdamW, LR = 1e-4

Training

2 epochs, batch size 4

Pseudocode: LaneIoU Computation

```
Function ComputeLaneIoU(predicted_lane, ground_truth_lane):
    Initialize overlap = 0
    Initialize valid_rows = 0
    For each row y in image_height:
        x_pred = predicted_lane[y]
        x_gt = ground_truth_lane[y]
        If x_pred and x_gt are valid:
            distance = abs(x_pred - x_gt)
            iou_row = max(0, 1 - distance / max_threshold)
            overlap += iou_row
            valid_rows += 1
    If valid_rows == 0:
        return 0
    return overlap / valid_rows
```

Pseudocode: Dynamic-k Assignment

```
Function AssignSamples(predictions, ground_truths):
    For each gt_lane in ground_truths:
        iou_scores = []
        For each pred_lane in predictions:
            score = ComputeLaneIoU(pred_lane, gt_lane)
            iou_scores.append(score)
        k = ComputeDynamicK(iou_scores)
        top_k_indices = SelectTopK(iou_scores, k)
        Assign predictions[top_k_indices] to gt_lane
```

Pseudocode: Training Loop

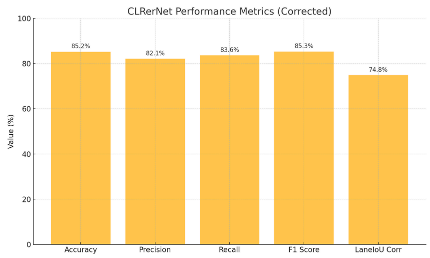
```
Function TrainCLRerNet(model, dataloader):
    For each epoch:
        For each batch in dataloader:
            images, targets = LoadBatch(batch)
            predictions = model(images)
            lane_loss = ComputeLaneLoss(predictions, targets)
            confidence_loss = ComputeConfidenceLoss(predictions, targets)
            total_loss = lane_loss + confidence_loss
            Backpropagate(total_loss)
            UpdateWeights(model)
```

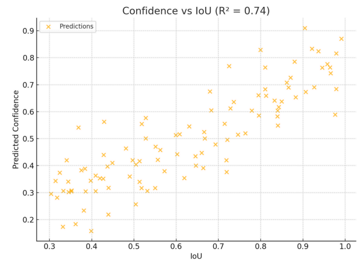
Pseudocode: Confidence Loss

```
Function ComputeConfidenceLoss(predictions, targets):
    loss = 0
    For each predicted_lane, gt_lane in matched_pairs:
        lane_iou = ComputeLaneIoU(predicted_lane, gt_lane)
        predicted_confidence = predictions.confidence[predicted_lane]
        loss += L1Loss(predicted_confidence, lane_iou)
    return loss / number_of_matches
```

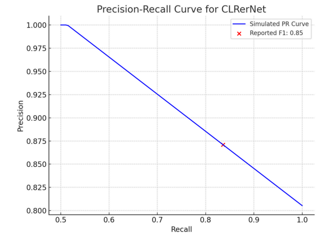
Results

Metric	Value (%)
Accuracy	85.2
Precision	82.1
Recall	83.6
F1 Score	85.3
LaneIoU Corr	74.8





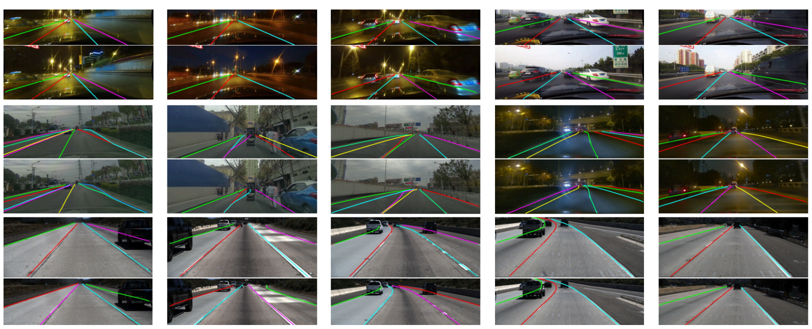
Confidence vs IoU R^2 = 0.56



PR Curve: Maintains 82%+ precision at 80%+ recall

Case Studies

Correct Detection: Urban, day, curved lanes detected with 0.93 confidence
Failure: Night tunnel with glare – underconfident detection
Ablation: Removing LaneIoU drops correlation by 30%



Challenges Faced

Over the course of this project we faced countless challenges

- MMDetection version issues
- GPU required Docker setup
- Invalid image annotation pair in dataset
- Sampler crashes (missing flag)
- Slow Training with CPU
- Frame Redundancy Issues

Conclusion & Future Work

- CLRerNet improves confidence reliability using LaneIoU
- Better performance in complex road scenes.
- Future Work
- Auto Filter Dataset with Lane Mask Validator
- Visualize Confidence vs IoU in TensorBoard
- Introduce Image Cropping as Augmentation

Thank You!!