A WEIGHTED BRANCH AGGREGATION BASED DEEP LEARNING MODEL FOR TRACK DETECTION IN AUTONOMOUS RACING

Shreya Ghosh*, Yi-Huan Chen*, Ching-Hsiang Huang*, Abu Shafin Mohammad Mahdee Jameel*, Aly El Gamal* & Samuel Labi†
Department of Electrical and Computer Engineering*, Department of Civil Engineering†,
Purdue University,
West Lafayette, IN 47907, USA.
{ghosh64,chen3966,huan1715,amahdeej,elgamala,labi}@purdue.edu

ABSTRACT

Intelligent track detection is a vital component of autonomous racing cars. We develop a novel Weighted Branch Aggregation based Convolutional Neural Network (WeBACNN) model that can accurately detect the track while being robust against image blurring due to high speed, and can work independently of lane markings. The code and dataset for this work is available at https://github.com/ghosh64/track-detection.

1 INTRODUCTION

Autonomous cars rely heavily on camera-based perception algorithms to perform navigation tasks like track detection. While research has mostly focused on lane detection in city traffic setting, track detection in autonomous racing applications pose some different challenges. Fast speed of cars on race tracks make images captured on the cameras prone to blurring, making it hard to distinguish the boundaries. State-of-the-art lane detectors rely heavily on lane boundary markings to make predictions, however they may not be present on a race track. Real time processing in autonomous racing cars require fast and efficient models, as onboard processors have to process information from several sensors at the same time. It is also important to accurately identify the area of the track in front of the vehicle as that is the actual space the vehicle can move into, defined as the Field of Perception (FoP). In this paper, we develop a fast lane detection technique that can overcome the effects of motion blurring, achieve superior FoP and track detection performance even in the absence of guiding lines.

2 EXISTING WORK

There are three types of approaches to lane detection in the current literature: parameter based methods, anchor based methods, and segmentation based methods. Parameter based methods include approaches like curve fitting using polynomials as seen in Feng et al. (2022); Torres et al. (2020). Anchor based methods would include Line-CNN(Li et al. (2020)) and LaneAtt(Tabelini et al. (2021)). Both methods use suggested lane lines. One uses a CNN and the other uses an attention based mechanism. Segmentation based methods treat lane detection as a segmentation task.Zheng et al. (2022) combine both high level and low level features in their model by including global context refined by low level features. Zhang et al. (2021) propose a multi-level memory aggregation network for this task. Wang et al. (2022) suggest Global Association Network(GANet) where each keypoint is directly regressed to the starting point of the lane and propose Lane-aware Feature Aggregator that supplements local information with global association.

3 MODEL ARCHITECTURE

Figure 1 shows the model architecture of WeBACNN. WeBACNN has a module for predicting the FoP, and another for estimating the track masks. The track mask network contains two branches - global and local. The global branch utilizes larger kernel sizes and strides that introduce spatial invariance to translation. The local feature branch has smaller convolution and pooling kernels, allowing us to extract localized details in a region of the image. The novel Weighted Branch Aggregation algorithm first crops the image into top, middle and bottom sections, and then uses context-aware

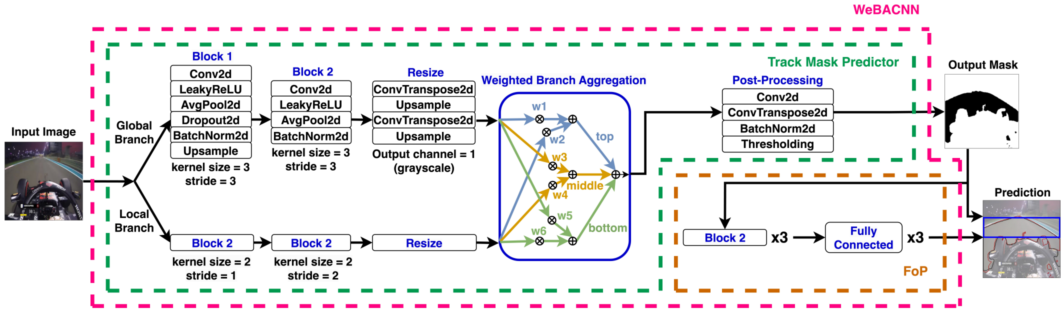

Figure 1: Architecture of the Proposed Model

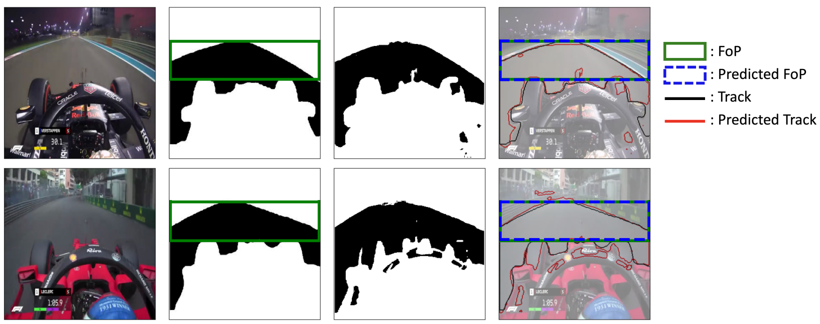

Figure 2: From Left to Right - Input, Ground Truth, Predicted Track Mask, Final Output with IoU.

data-driven weights on each of the sections for aggregating the global branch and the local branch. Firstly, in the top section, the global branch is weighted heavier because higher level features for a global context are more important for capturing a clearer look-ahead image. Then for the middle section, the local features are weighted slightly more to ensure smooth reconstruction for curves that comes from lower level features as well as context from higher level features containing spatial information. Finally, the local branch is weighted heavier in the lower section as the track prediction needs to accommodate the irregularities of the race car's shape. In the post-processing step, the kernel sizes are kept small to keep most part of the prediction unchanged while smoothing the detection for the edges. Then the FoP predictor is employed to obtain the final output. The weights [w1,w2,w3,w4,w5,w6] are [0.7,0.3,0.4,0.6,0.3,0.7], based on empirical experiments.

4 Results and Future Work

There are no autonomous racing datasets specifically containing annotated track boundary data to the best of our knowledge. We developed a distinct dataset for this specific task, whose details are presented in A.1. Since both our true labels and predicted labels are segmentation masks, we compute and present pixel-wise Intersection over Union (IoU) (A.2). Some preliminary results containing the IoU within the FoP, along with overall IoU is presented in Table 1,where we can see that the proposed weighting mechanism provides superior performance as compared to the same model with a non-weighted aggregation mechanism. In A.1, we present results with a detailed breakdown of the test dataset. WeBACNN is fast and more lightweight compared to the popular YOLOv8 model as elaborated in A.3.

Table 1: IoU for whole image and FoP

Method	IoU	IoU (FoP)
Proposed Method Without Weighted Aggregation	0.6188	0.7671
WeBACNN	**0.6811**	**0.7899**

In Figure 2, we present two examples (with and without lane markings) of the original input image, true label with true FoP annotations, predicted track mask, and the output containing the predicted track mask and predicted FoP area as well as the true label and true FoP annotations. It can be seen from Figure 2 that even when the input images have motion blurring or no bounding lines, our model is still able to accurately predict the track. Moving forward, our focus will be on improving detection of the vehicle outline, scenarios involving competing cars, and comparison with more track detection algorithms.

URM Statement

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers Track.

References

Zhengyang Feng, Shaohua Guo, Xin Tan, Ke Xu, Min Wang, and Lizhuang Ma. Rethinking efficient lane detection via curve modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17062–17070, June 2022.

Xiang Li, Jun Li, Xiaolin Hu, and Jian Yang. Line-cnn: End-to-end traffic line detection with line proposal unit. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):248–258, 2020. doi: 10.1109/TITS.2019.2890870.

Lucas Tabelini, Rodrigo Berriel, Thiago M. Paixao, Claudine Badue, Alberto F. De Souza, and Thiago Oliveira-Santos. Keep your eyes on the lane: Real-time attention-guided lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 294–302, June 2021.

Lucas Tabelini Torres, Rodrigo Ferreira Berriel, Thiago M. Paixão, Claudine Badue, Alberto F. De Souza, and Thiago Oliveira-Santos. Polylanenet: Lane estimation via deep polynomial regression. *CoRR*, abs/2004.10924, 2020. URL `https://arxiv.org/abs/2004.10924`.

Jinsheng Wang, Yinchao Ma, Shaofei Huang, Tianrui Hui, Fei Wang, Chen Qian, and Tianzhu Zhang. A keypoint-based global association network for lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1392–1401, June 2022.

Yujun Zhang, Lei Zhu, Wei Feng, Huazhu Fu, Mingqian Wang, Qingxia Li, Cheng Li, and Song Wang. VIL-100: A new dataset and A baseline model for video instance lane detection. *CoRR*, abs/2108.08482, 2021. URL `https://arxiv.org/abs/2108.08482`.

Tu Zheng, Yifei Huang, Yang Liu, Wenjian Tang, Zheng Yang, Deng Cai, and Xiaofei He. Clrnet: Cross layer refinement network for lane detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 898–907, June 2022.

A Appendix

A.1 Additional Results

Table 2: Test Dataset Composition

	Total	Day	Night	Blurry	With Lane	W/o Lane	Walls	Grass Banks
No. of Images	63	48	15	10	54	9	24	23

The dataset is developed by capturing frame-by-frame images from online racing videos, and then annotating them with track and FoP data. We utilize a pretrained YOLOv8 segmentation model augmented with manual annotation to establish the ground truths. Images captured on a fast moving autonomous vehicle are often blurry, lack bounding lines for lanes and are surrounded by different environments. This dataset contains examples from each of these categories, making our model robust against the challenges of racing data. In Table 2, we show an overview of the composition of the test dataset. A single image may fall in more than one category. The entire dataset contains 331 images and labels. 80% of this dataset is used during training and 20% is used for testing.

Then in Figure 3, we show some of these diverse conditions in the data such as being surrounded by grass or walls, day time or night time, with or without lane markings and blurring. Table 3 shows the performance of our model in terms of the IoU and IoU(FoP) for these select cases. As it can be seen, our model is robust against the challenges present in these cases.

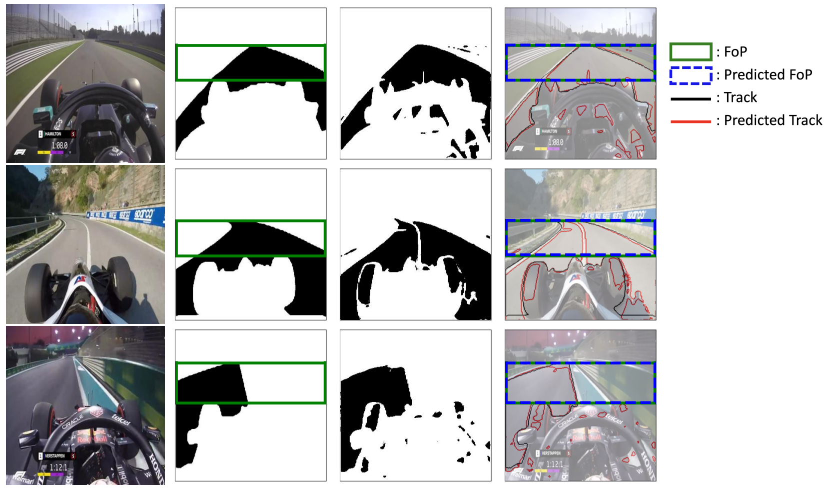

Figure 3: Some examples of diverse scenarios: From Left to Right - (1) Input, (2) Ground Truth, (3) Predicted Track Mask, (4) Final Output with IoU. From Top to Bottom - (1) Surrounded by grass, no lane markings, day; (2) Surrounded by walls, with lane markings, day; (3) Surrounded by grass, no lane markings, blurred, night.

Table 3: IoU and IoU(FoP) for select cases

Scenario	**IoU**	**IoU (FoP)**
Grass banks, day	0.7441	0.9320
Walls, with lane markings, day	0.7403	0.8667
No lane markings, blurred, night	0.7782	0.9338

A.2 Pixel-wise IoU Computation

We define the pixel-wise IoU as follows:

$$IoU(prediction, label) = \frac{track\_pixels(prediction \cap label)}{track\_pixels(prediction \cup label)} \tag{1}$$

Where $track\_pixels$ gives us the number of pixels that meet the required condition. In this case, the intersection represents the number of pixels that are marked as lane both in the prediction and the label. The union represents total number of pixels that are marked as lane in the prediction or in the label. For the IoU (FoP) metric, we estimate the track over the area bounded by the predicted FoP, and compare it against the ground truth track area, bounded by the ground truth FoP.

A.3 Complexity

Table 4: YOLOv8 VS Proposed Model: Inference Time, FLOPS and Parameters

Metrics	**YOLOv8**	**WeBACNN**
Inference Time(ms)	12.3	**0.26**
FLOPs(G)	12.1	**7.37**
Parameters(M)	3.26	**1.15**

In Table 4, we present metrics (Inference time for a single frame, Floating Point Operations Per Second (FLOPs), and number of parameters) to compare the complexity of the YOLOv8 segmentation model with our proposed method. In a real world scenario, this algorithm would likely be deployed on a resource constrained device. Ideally the model should have a lower memory requirement and a fast inference time to perform real time track detection. Our model is able to address this requirement better than the YOLOv8 model.