TinyML-Empowered Line Following for a Car Robot

Evaluating the Capabilities of Various Lane Detection Models on Microcontrollers

AUTHOR Adrien Carton
RESPONSIBLE PROFESSOR Dr. Qing Wang
SUPERVISOR Ran Zhu

TU Delft

Introduction:

- Existing methods accurate up to 98%
- Hyperfocus on accuracy, resource constraints not considered
- Consequence: GPU needed for near unanimity of models
- Models are inadequate for release on full scale and small scale vehicles without use of cloud or large hardware

Our question: *What is the best solution, focusing on efficiency and accuracy, to detect lanes dynamically on a lightweight microcontroller?*

- Dynamically implies real time performance
- Microcontroller: Raspberry Pi Pico 0 used as most simple and constrained medium with 264KB SRAM

Background:

Image Processing (IP) approaches:

- IP pipelines with up to 97% accuracy
- Efficient in open spaces with clear lanes
- Struggle with smaller context
- Methods from IP combined with ML in research
- Effect of different down sampling processes and edge filtering as a preprocessing method is explored

Machine Learning (ML) approaches:

- CNN revolution [1], two models combined with layer for edge detection and layer to localize lanes
- U-Net [2]: CNN with “skip connections”, previous convolution results reused in deconvolution, edges remain very sharp
- UFLD [3]: classification task instead of segmentation, for each height h the lane is defined as two labels x1 and x2
- SCNN [4]: propagates information in columns and rows, gains spatial awareness

All models are variants of CNN

Obfuscated lanes and more complex situations recognized

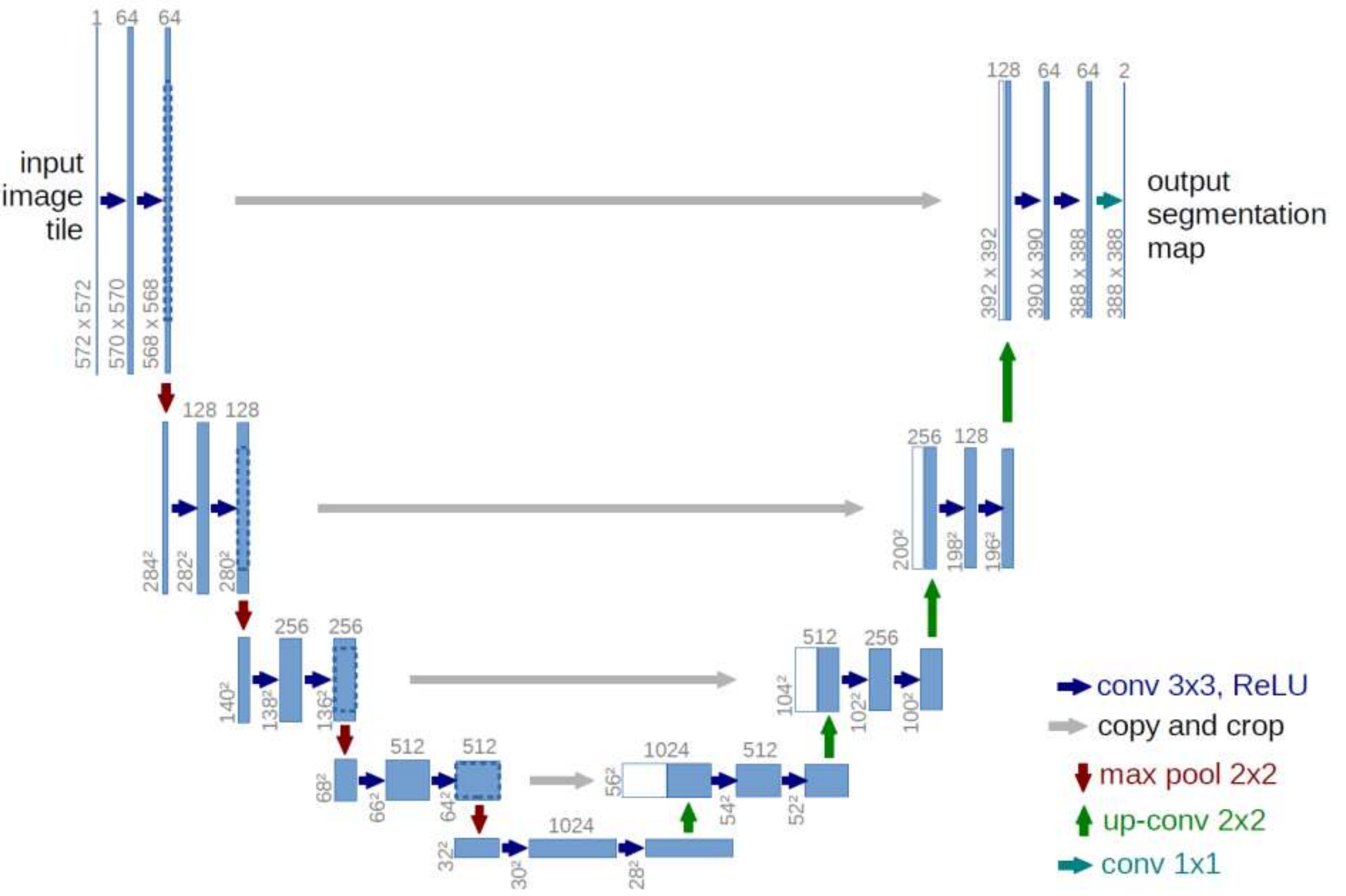

Figure 1: U-Net architecture [3] with skip connections, the U shape of the process helps keep edges and baptized the models

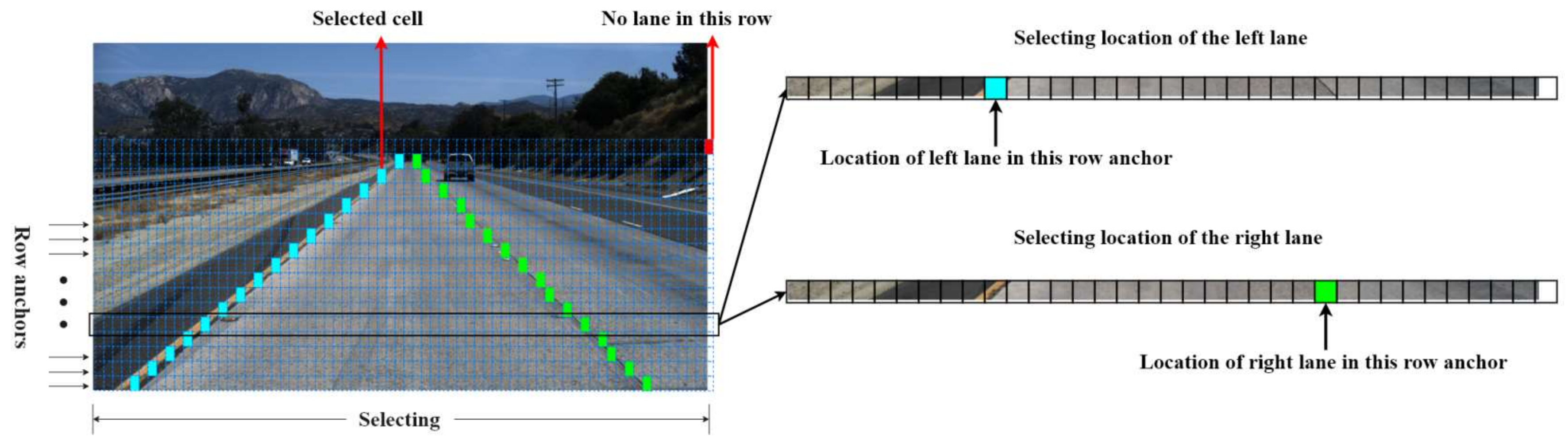

Figure 2: UFLD [4] classification task, each anchor row has as classes possible coordinates for the lane

Experimentation:

- 2 architectures with adequate accuracy run on Pico 0
- UFLD and U-Net
- U-Net boasts larger Accuracy: 95%
- UFLD fastest inference speed: 88ms per frame
- Narrower U-net trained to increase speed but accuracy loss was too large to justify time gain
- Preprocessing had little cost and little benefit

Model Type	Inference time (ms)	Tensor Arena (KB)
UFLD 40	81	21.3
Mini U-Net 40	650	67.5
U-Net 40	721	79.8
UFLD 80	220	40.0
Mini U-Net 80	-	overflow
U-Net 80	-	overflow

Table 1: Result of inference on Raspberry Pi Pico 0 for all final trained models

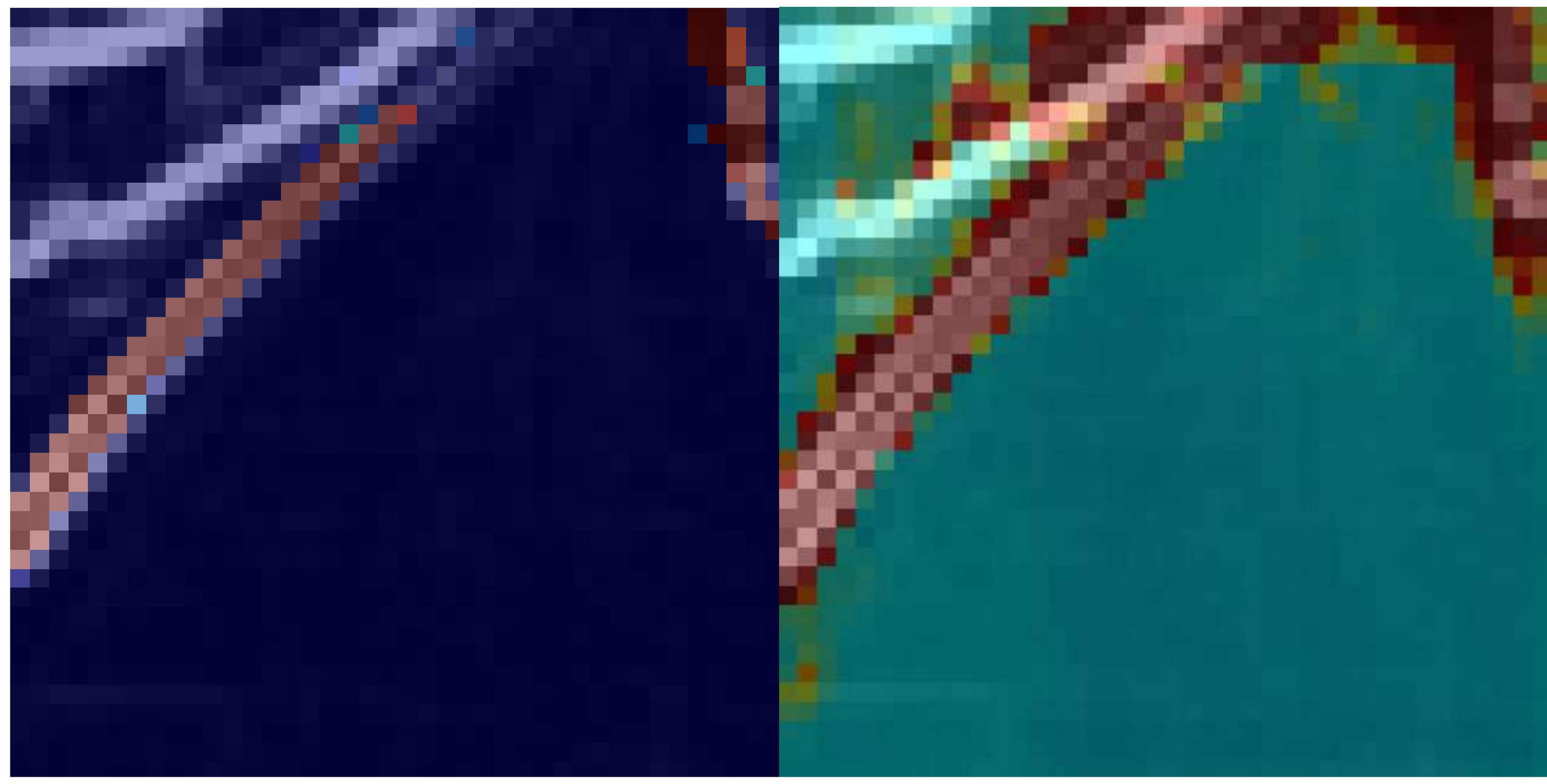

Figure 3: results of inference, Left wide U-Net with Dice Loss, Right narrower U-net with Dice Loss and Focal Loss to encourage guesses Images from the TUSimple dataset [5] blurred and cropped to 40x40 pixels

Limitations:

-Model is only able to infer on simple linear inputs

- Models were restricted to an extremely small input size of 40x40 or 80x80 pixels

Conclusions:

- “Best” model is debatable:
  - UFLD offers real time performance on Pico 0
  - U-Net offers more precise predictions, accuracy of 95% on 40x40 input, with more detail
- Effect of input image preprocessing is much smaller than expected

Future works:

- Increasing library of architectures compatible with microcontrollers
- Training models for larger microcontrollers with larger inputs
- Switching feature space from point of view to bird’s eye view

References:

[1] Z. Wang, W. Ren, and Q. Qiu, “LaneNet: real-time lane-detection networks for autonomous driving,” 2018.
[2] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: convolutional networks for biomedical image segmentation,” 2015
[3] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, “Spatial as deep: Spatial CNN for traffic-scene understanding,” 2018
[4] Z. Qin, H. Wang, and X. Li, “Ultra-fast structure-aware deep lane detection,” in Proc. ECCV, Glasgow, U.K., Aug. 2020
[5] F. Pizzati, M. Allodi, A. Barrera, and F. García, “Lane detection and classification using cascaded CNNs,” 2019