Predicting Motion of Vulnerable Road Users using High-Definition Maps and Efficient ConvNets

Fang-Chieh Chou, Tsung-Han Lin, Henggang Cui, Vladan Radosavljevic,
Thi Nguyen, Tzu-Kuo Huang, Matthew Niedoba, Jeff Schneider, Nemanja Djuric

Uber ATG

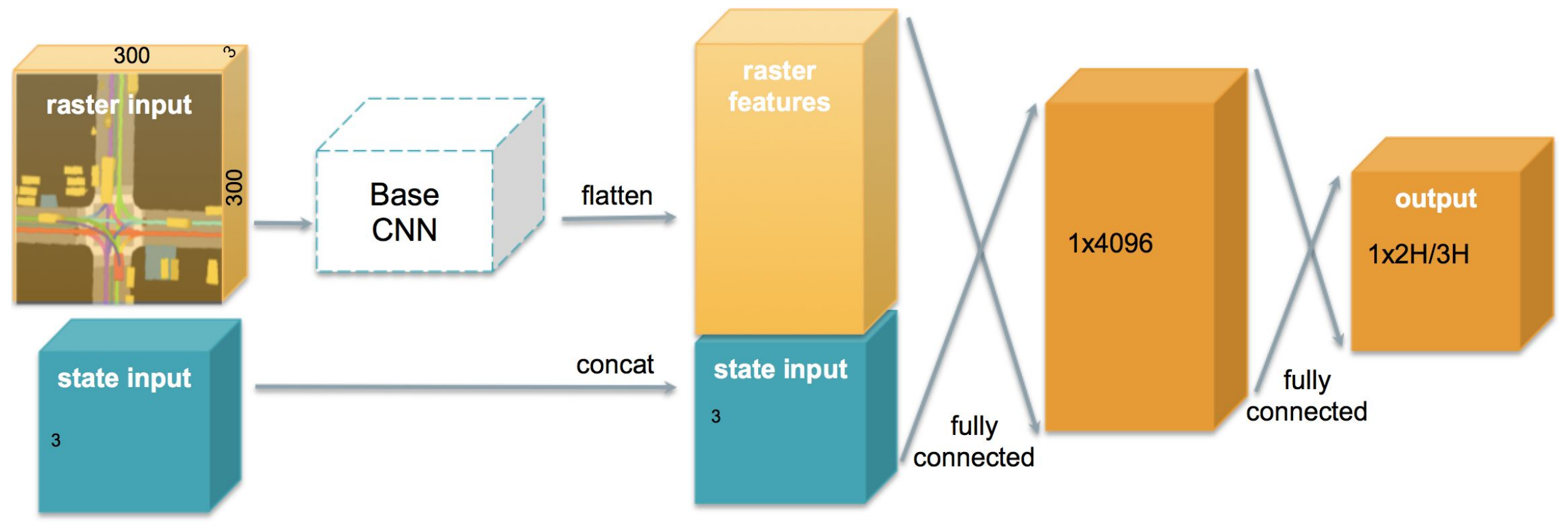
Abstract

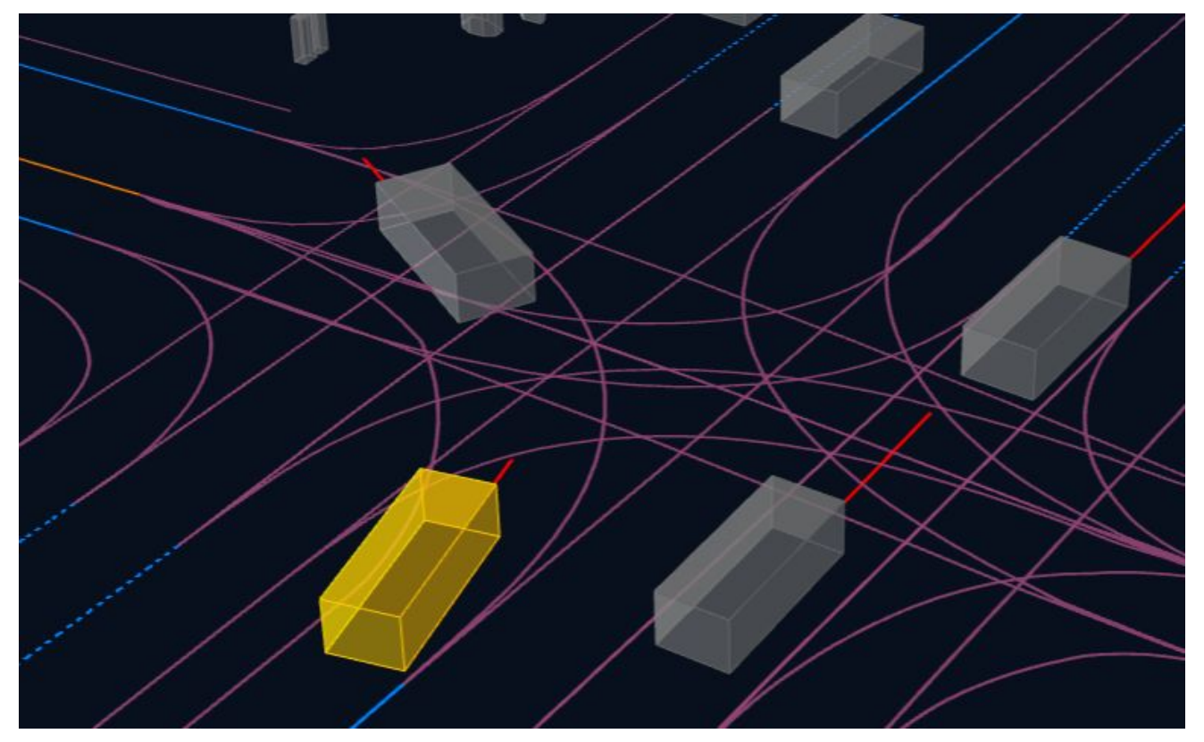
Following detection and tracking of traffic actors, prediction of their future motion is the next critical component of a self-driving vehicle (SDV), allowing the SDV to move safely and efficiently in its environment. This is particularly important when it comes to vulnerable road users (VRUs), such as pedestrians and bicyclists. We present a deep learning method for predicting VRU movement where we rasterize high-definition maps and actor's surroundings into bird's-eye view image used as input to convolutional networks. In addition, we propose a fast architecture suitable for real-time inference, and present an ablation study of rasterization choices.

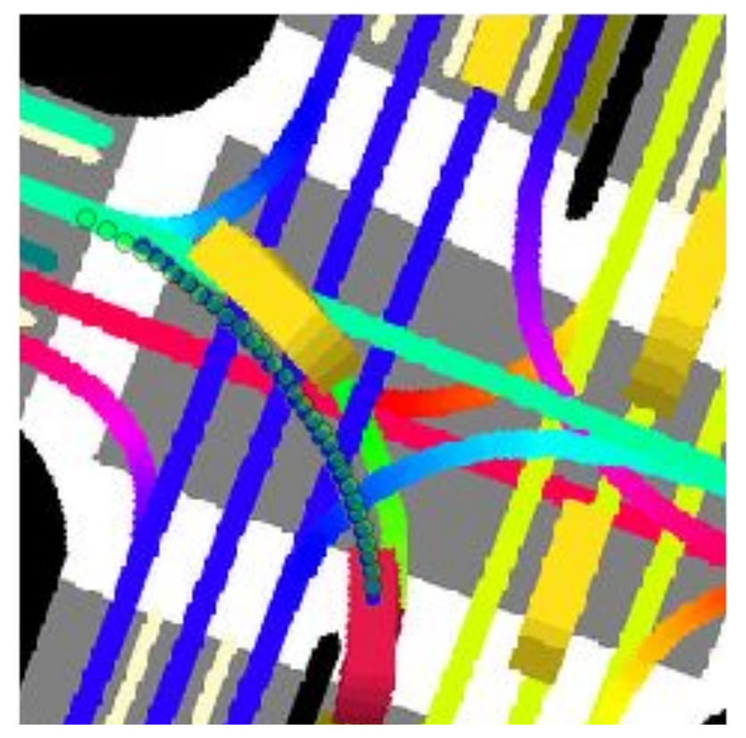
Prior Work (RasterNet)

We followed the same setup used in our previous work (originally applied to vehicle actors). In this paper we demonstrate that the methodology can be successfully applied to VRU actors.

We combined the output of an existing detection and tracking system (objects with state estimates, e.g. positions, bounding boxes, velocities), with HD map information (locations of lanes and crosswalks, lane directions, etc.) into an object-centric raster input image for each object. A CNN is then applied to predict the future positions of the target objects.







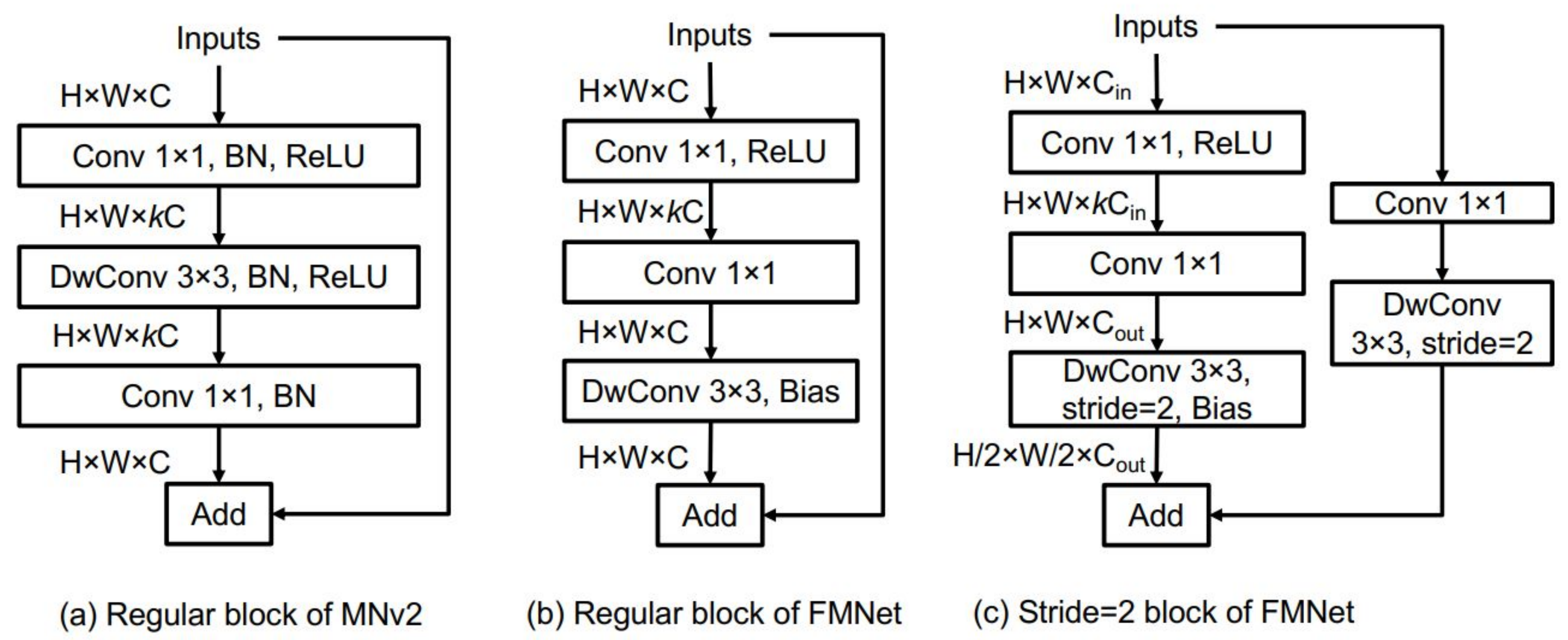
Reference

N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, and J. Schneider. Short-term motion prediction of traffic actors for autonomous driving using deep convolutional networks. *arXiv preprint arXiv:1808.05819, 2018*.

Methods & Results

This work consists of two parts. First we propose a fast CNN architecture suitable for running in real-time onboard the SDV. Secondly we present an ablation study of various rasterization settings.

Fast CNN architecture:
We performed simple modifications on the MobileNet-V2 architecture to speed up the inference. Operations with high memory access costs are removed or reordered.
The resulted architecture (FastMobileNet) is significantly faster on batched GPU inference (batch size = 32) than the original MobileNet-V2, with comparable accuracy.



(a) Regular block of MNv2 (b) Regular block of FMNet (c) Stride=2 block of FMNet

Figure 1: Building blocks of MobileNet-v2 and FastMobileNet

Layer	Output	Stride	Repeats
Raster image	$300 \times 300 \times 3$	–	–
Conv 3×3	$150 \times 150 \times 24$	2	1
DwConv 3×3	$75 \times 75 \times 24$	2	1
FMNet block 1	$75 \times 75 \times 12$	1	2
FMNet block 2	$38 \times 38 \times 16$	2	3
FMNet block 3	$19 \times 19 \times 32$	2	4
FMNet block 4	$19 \times 19 \times 48$	1	3
FMNet block 5	$10 \times 10 \times 80$	2	3
FMNet block 6	$10 \times 10 \times 160$	1	1
Conv 1×1	$10 \times 10 \times 640$	1	1
Global average pooling	$1 \times 1 \times 640$	1	1

Ablation Study:
We performed ablation studies on several rasterization choices of our input image representation.

- Raster frame rotation: whether we rotate the frame so that the object's heading point ro north.
- Raster pixel resolution: spatial size of the pixel; larger resolution covers larger context but loses fine details.
- Lane direction: lane direction is encoded as a rainbow color for each lane segment.
- Traffic light: traffic lights as colored circles at intersections; uncrossable crosswalks are also colored green.
- Learned colors: instead of using manually picked colors, we let the network learn the color of each layer.
- Model pretraining: instead of training from scratch, we initialize VRU model with model trained on vehicle data.

Architecture	Pred. error [m]	Latency [ms]	FLOPs	Num. parameters
AlexNet	1.36	15.8	2.84G	70.3M
ResNet18	1.29	36.2	6.80G	11.7M
MNv2-0.5	**1.27**	21.3	**322M**	581k
MnasNet-0.5	1.28	18.3	335M	825k
FMNet	1.28	**12.1**	363M	**564k**

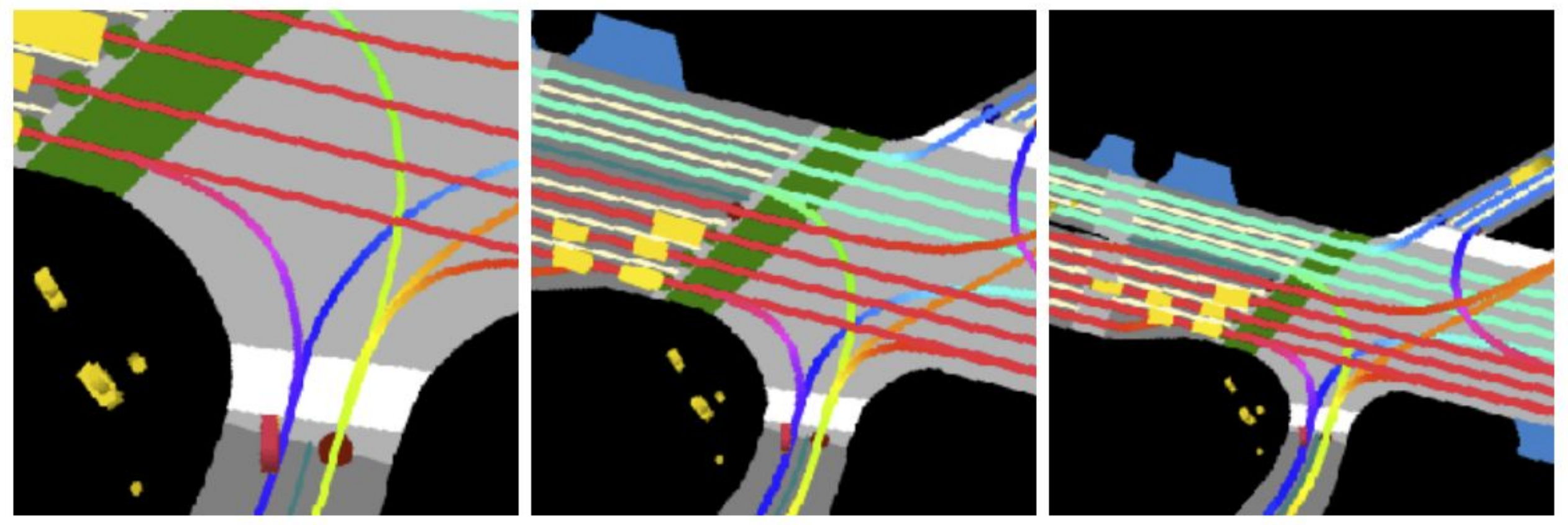


Figure 2: Raster images for bicyclist actor (colored red) using resolution of 0.1m, 0.2m, and 0.3m

		Bicyclists			Pedestrians		
Approach	Resolution	Average	@1s	@5s	Average	@1s	@5s
UKF	–	2.89	0.80	6.60	0.67	0.22	1.22
RasterNet	$0.1m$	1.07	0.43	2.73	**0.51**	**0.17**	**0.90**
RasterNet	$0.2m$	1.07	0.44	2.72	0.52	0.18	0.93
RasterNet	$0.3m$	1.09	0.45	2.80	0.53	0.18	0.95
RasterNet w/o rotation	$0.2m$	1.29	0.49	3.30	0.58	0.20	1.02
RasterNet w/o traffic lights	$0.2m$	1.11	0.44	2.86	0.55	0.20	0.96
RasterNet w/o lane headings	$0.2m$	1.07	0.43	2.72	0.52	0.18	0.93
RasterNet with learned colors	$0.2m$	**1.05**	**0.42**	**2.70**	0.53	0.18	0.93
RasterNet with car pretraining	$0.2m$	**1.05**	**0.42**	**2.70**	0.59	0.20	1.05

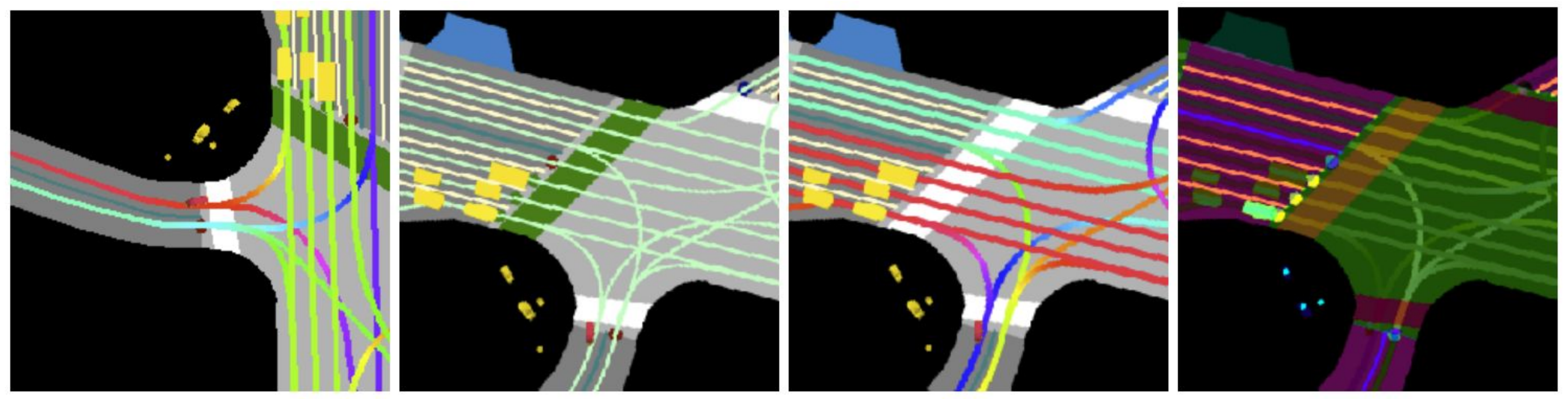


Figure 3: Different rasterization settings with 0.2m resolution for bicyclist example: (a) no rotation, (b) no lane heading, (c) no traffic light, (d) learned colors

Conclusion

We presented an efficient and effective solution to motion prediction of VRU actors. This is a critical problem in autonomous driving, as such actors have higher risk of injury and are less predictable since they may change behavior faster than vehicles. We applied recently proposed rasterization technique to generate raster images of actors' surroundings encoding their context, used as input to deep CNN trained to predict actor trajectory. Moreover, we proposed a fast architecture suitable for real-time operations, and finally presented a detailed ablation study of various rasterization choices.