State Space Models for Event Cameras

Nikola Zubić, Mathias Gehrig, Davide Scaramuzza

University of Zurich[UZH]

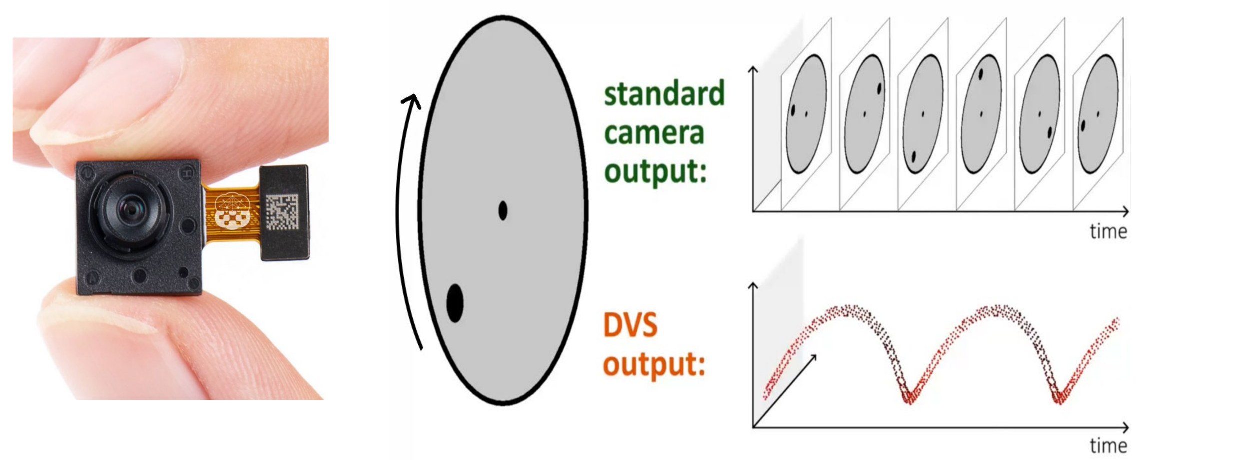
ROBOTICS & PERCEPTION GROUP

Event-based Vision Research

Motivation:

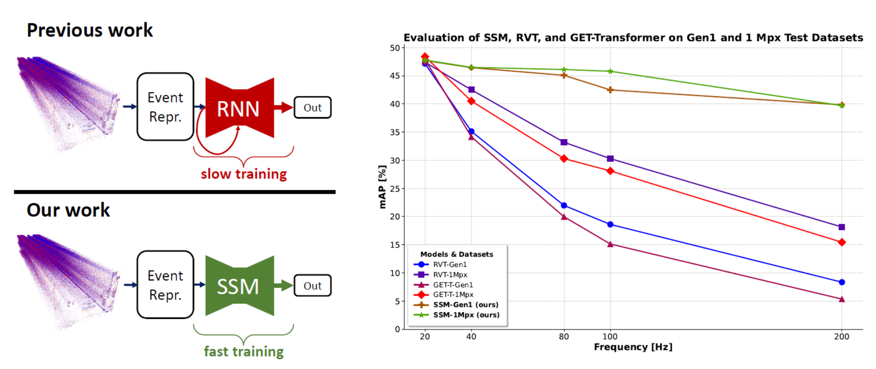
Motivation: State-of-the-art event camera algorithms **struggle** with training and inference **speeds** and **lack** adaptability across different (higher) **frequencies**. We introduce State Space Models (**SSMs**) to Event Cameras to **solve all** of these **issues**.

What is an event camera?



- Only transmits **brightness changes**
- Output is a stream of **asynchronous events**
- **Advantages:** low latency, no motion blur, HDR

Check out our Paper and Video!



Code & Video

Sponsors

Horizon2020 European Union Funding for Research & Innovation

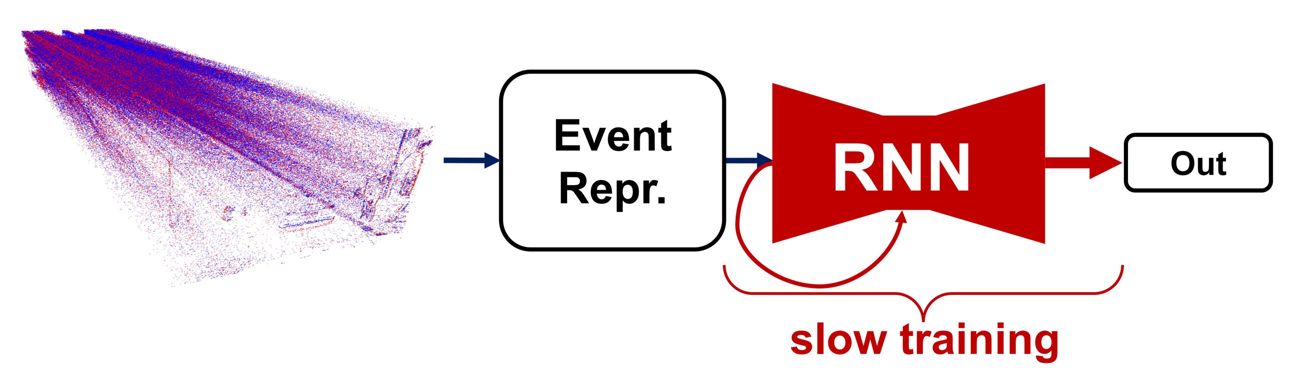
erc

FNSNF Swiss National Science Foundation

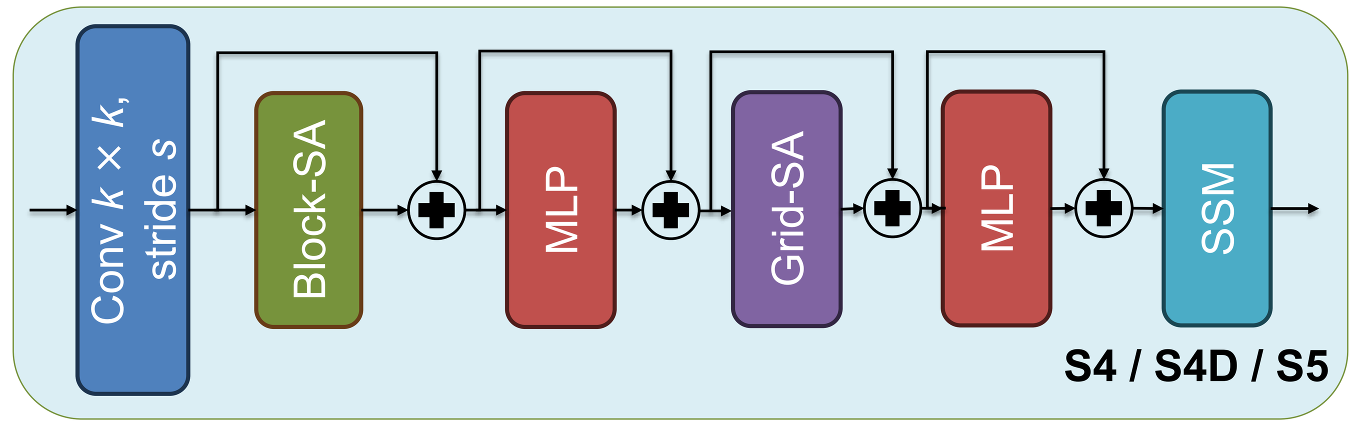
robotics Swiss National Centre of Competence in Research

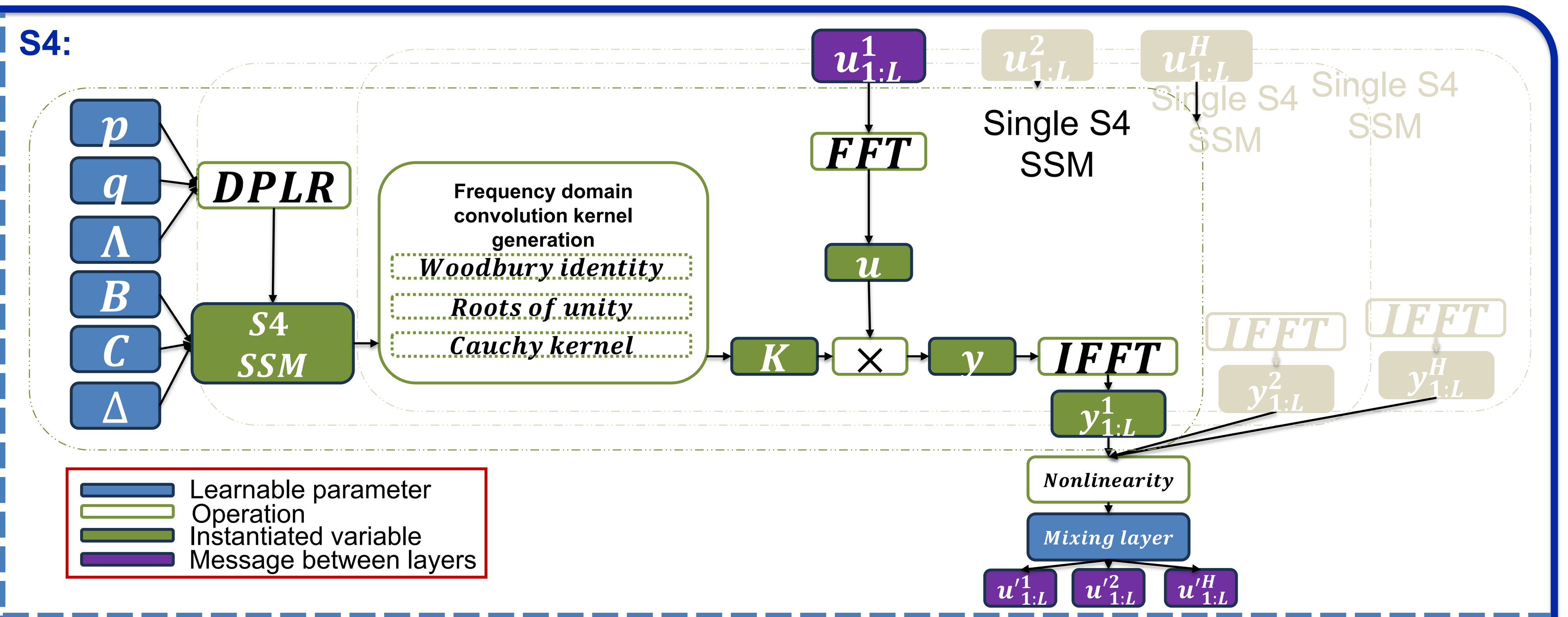
Method Overview:

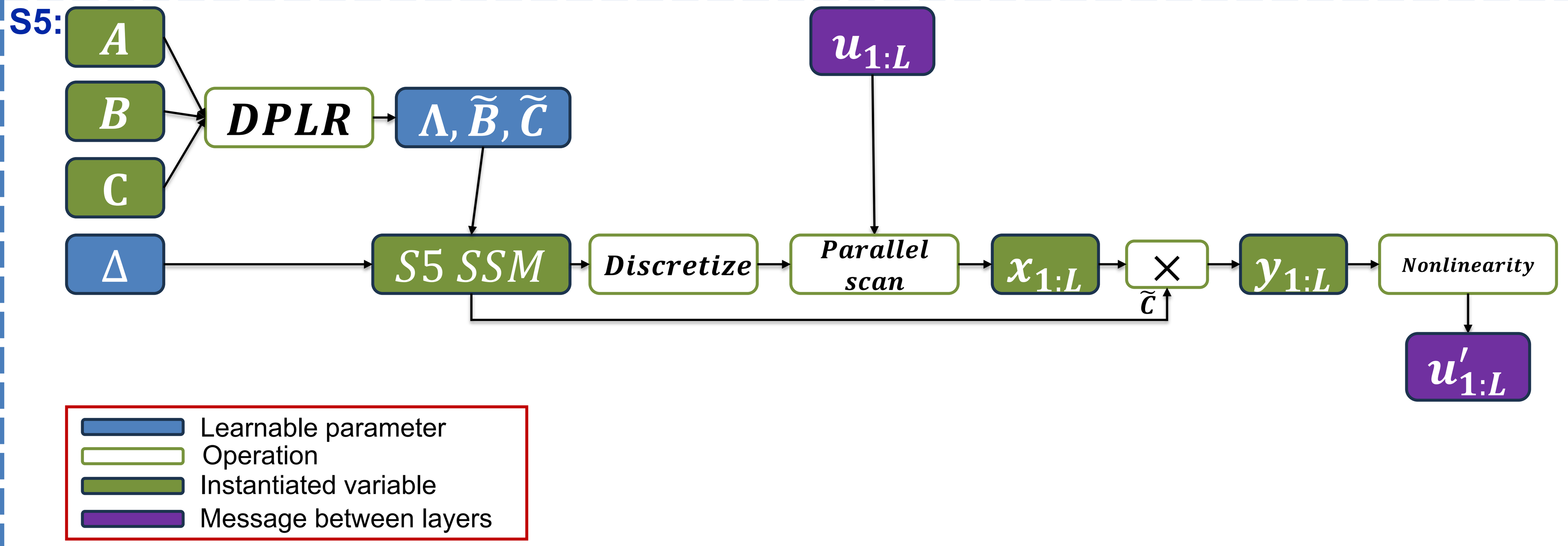
Previous work



Our work







Ablation results:

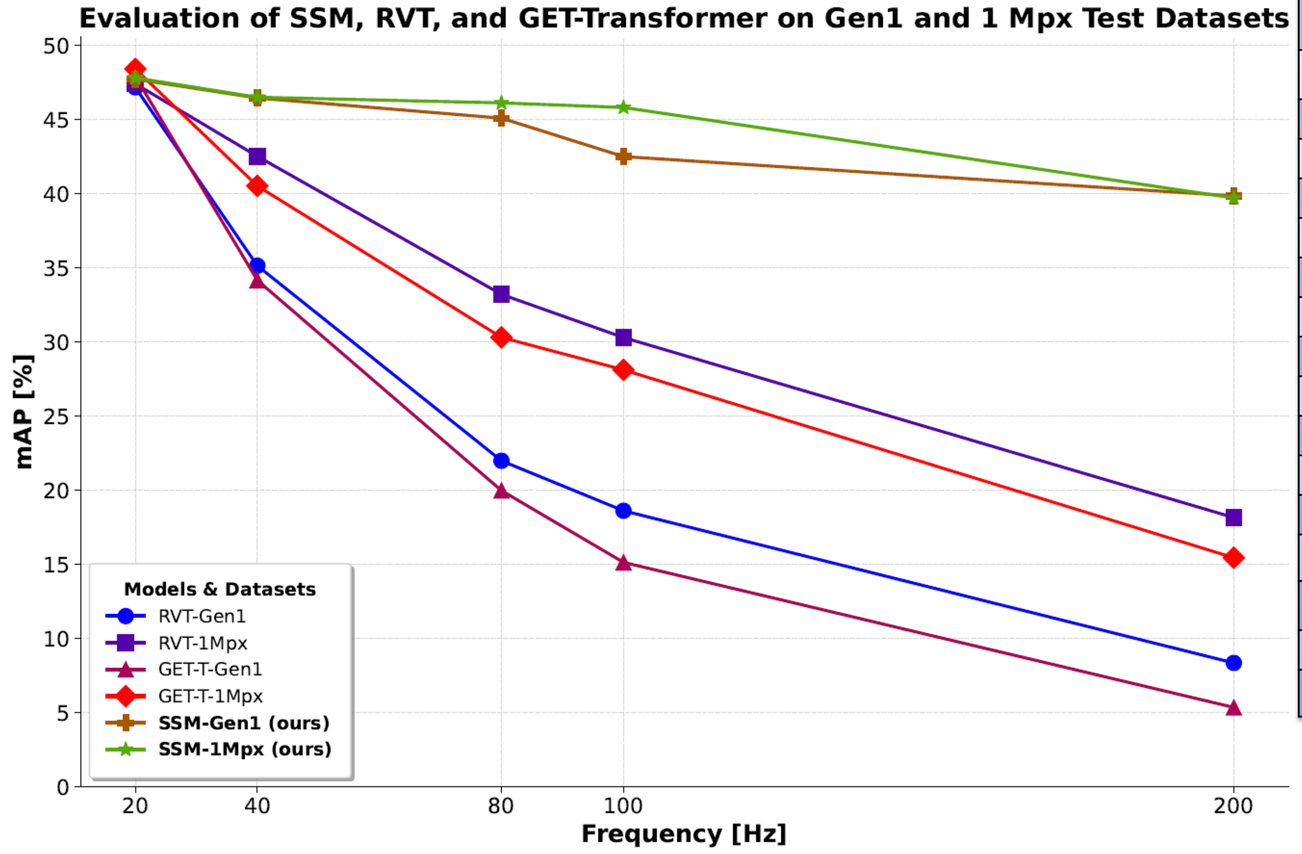
Model	α = 0	α = 0.5	α = 1	Average
S4-legS	46.66	-	-	46.66
S4D-legS	46.93	47.33	46.50	46.92
S4D-inv	46.15	46.23	46.11	46.16
S4D-lin	44.82	46.02	45.04	45.29
S5-legS	48.33	**48.48**	48.00	**48.27**
S5-inv	47.26	<u>47.43</u>	46.98	<u>47.22</u>
S5-lin	46.12	46.40	45.59	46.04

Performance comparison between the **S4, S4D and S5 models** for different values of **α** and initializations (**legS, inverse, linear**) on Gen1 validation dataset. α = 1.0 corresponds to Nyquist limit.

S1	S2	S3	S4	mAP_{RVT}	mAP_{S4D}	mAP_{S5}
				33.90	39.99	43.67
			✓	41.68	43.11	46.10
		✓	✓	46.10	45.33	47.52
	✓	✓	✓	48.82	47.02	48.41
✓	✓	✓	✓	49.52	47.33	48.48

SSM contribution in various stages on the Gen1 dataset. S4D and S5 use α = 0.5.

Main results:



Method	Backbone	Detection Head	Gen1 mAP	Gen1 Time (ms)	1 Mpx mAP	1 Mpx Time (ms)	Params (M)
Asynet	Sparse CNN	YOLOv1	14.5	-	-	-	11.4
AEGNN	GNN	YOLOv1	16.3	-	-	-	20.0
Spiking DenseNet	SNN	SSD	18.9	-	-	-	8.2
Inception + SSD	CNN	SSD	30.1	19.4	34.0	45.2	> 60*
RRC-Events	CNN	YOLOv3	30.7	21.5	34.3	46.4	> 100*
MatrixLSTM	RNN + CNN	YOLOv3	31.0	-	-	-	61.5
YOLOv3 Events	CNN	YOLOv3	31.2	22.3	34.6	49.4	> 60*
RED	CNN + RNN	SSD	40.0	16.7	43.0	39.3	24.1
ERGO-12	Transformer	YOLOv6	**50.4**	69.9	40.6	100.0	59.6
RVT-B	Transformer + RNN	YOLOX	47.2	10.2	47.4	11.9	18.5
Swin-T v2	Transformer + RNN	YOLOX	45.5	26.6	46.4	34.5	21.1
Nested-T	Transformer + RNN	YOLOX	46.3	25.9	46.0	33.5	22.2
GET-T	Transformer + RNN	YOLOX	<u>47.9</u>	16.8	**48.4**	18.2	21.9
S4D-ViT-B (ours)	Transformer + SSM	YOLOX	46.2	9.40	46.8	10.9	16.5
S5-ViT-B (ours)	Transformer + SSM	YOLOX	47.7	<u>8.16</u>	<u>47.8</u>	<u>9.57</u>	18.2
S5-ViT-S (ours)	Transformer + SSM	YOLOX	46.6	**7.81**	46.5	**8.87**	9.7

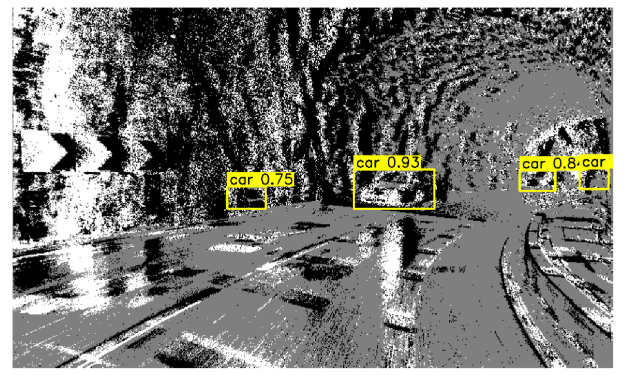
Comparisons on test sets of Gen1 and 1 Mpx datasets (20 Hz). Best results in **bold** and second best underlined. A star * suggests that this information was not directly available and estimated based on the publications. Runtime is measured in milliseconds for a batch size of 1. We used a T4 GPU for SSM-ViT to compare against indicated timings in prior work on comparable GPUs (Titan Xp). Our model is the **3rd best on the Gen1** and **2nd best on the 1Mpx** dataset in terms of downstream task performance, while having **fastest inference and less parameters**.

Model	Dataset	20 Hz	40 Hz	80 Hz	100 Hz	200 Hz	Perf. Drop
RVT	Gen1	47.16	35.13	21.98	18.61	8.35	26.14
	1Mpx	47.40	42.51	33.20	30.29	18.15	16.36
S5	Gen1	47.71	46.44	45.08	42.49	39.84	<u>4.25</u>
	1Mpx	47.80	46.49	46.11	45.80	39.70	**3.27**
GET	Gen1	47.90	34.15	19.97	15.13	5.35	29.25
	1Mpx	48.40	40.51	30.30	28.11	15.44	19.81

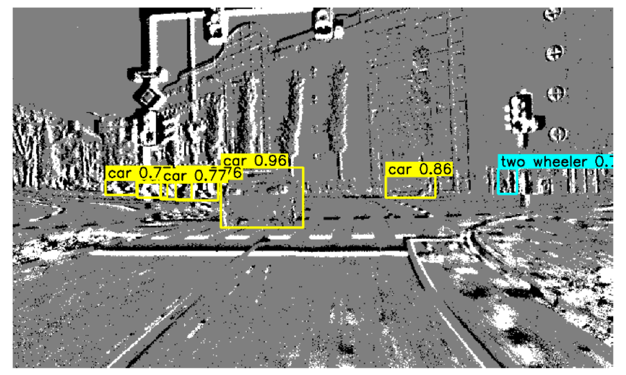
Evaluation of RVT, S5, and GET across different frequencies on test datasets. The Performance Drop is calculated as the average difference between the original performance at 20 Hz and performances at higher frequencies.

Model	mAP_{Gen1}	mAP_{1Mpx}
S5-ViT-B	**47.71**	**47.80**
S5-ConvNext-B	45.92	45.66
S5-SSM2D-B	46.10	45.74

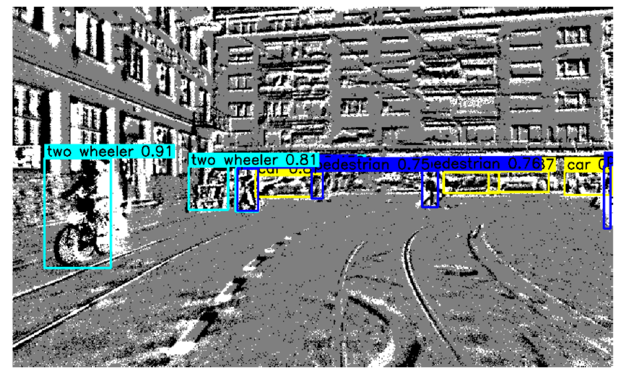
Comparison of mAP scores for Gen1 and 1 Mpx datasets across different base models.



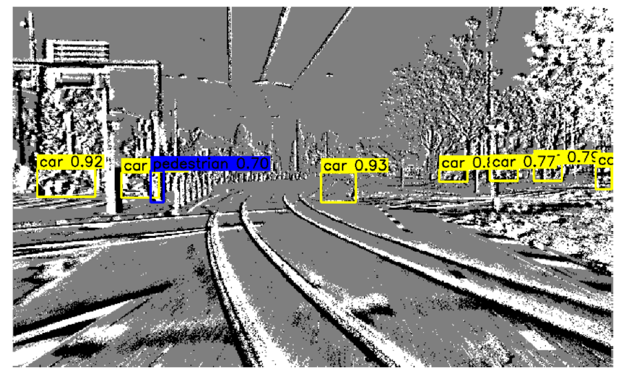
(a) *interlaken_00_b*



(b) *zurich_city_00_a*



(c) *zurich_city_00_b*



(d) *zurich_city_07_a*