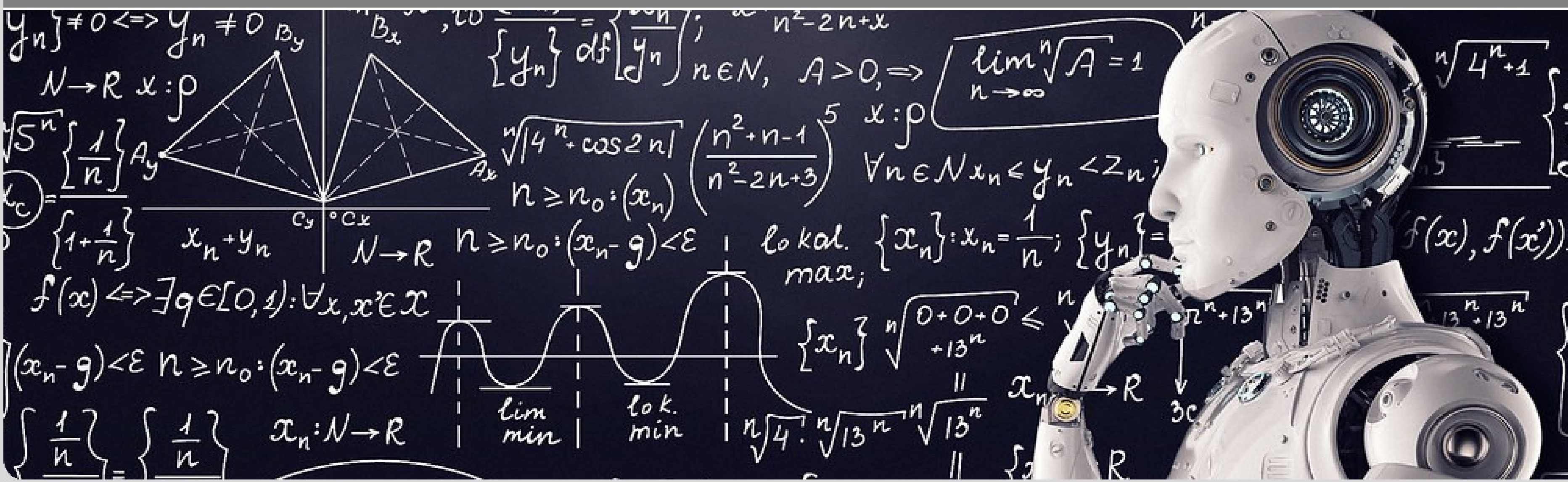
Karlsruhe Institute of Technology

UAV-Net: A Fast Aerial Vehicle Detector for Mobile Platforms

3rd International Workshop on Computer Vision for UAVs – CVPR 2019

Tobias Ringwald, Lars Sommer, Arne Schumann, Jürgen Beyerer and Rainer Stiefelhagen

Karlsruhe Institute of Technology – Institute for Anthropomatics and Robotics, Computer Vision for Human-Computer Interaction
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation



Motivation

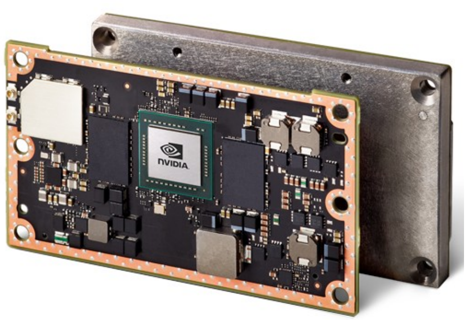
- Deep learning best solution for object detection
- Large server clusters for training and inference
- „Intelligence“ also desired in edge devices
- Problems with weight, power supply and dimensions

+

= ?

Solution

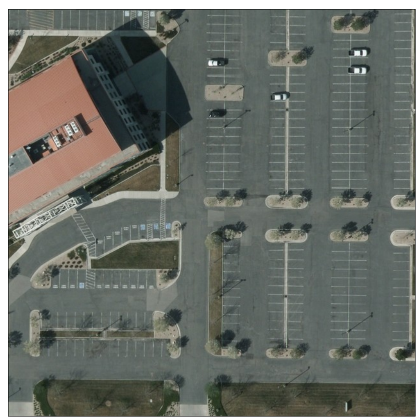
- Jetson platform by NVIDIA
- For use in „intelligent“ cars, cameras, **drones** etc.
- Embedded GPU with cuDNN stack
- Jetson TX2:
  - 8GB RAM
  - 6-core CPU @ 2GHz
  - 256 CUDA cores
  - Max. 15W
- Is it enough?



UAV-Net

- Small and efficient detector for on-board object detection
- Very low memory footprint
- On par with state-of-the-art detection models
- Evaluated on 3 different datasets
- Design decisions: **Meta-architecture, backbone, layers, filters**

DLR 3K Munich

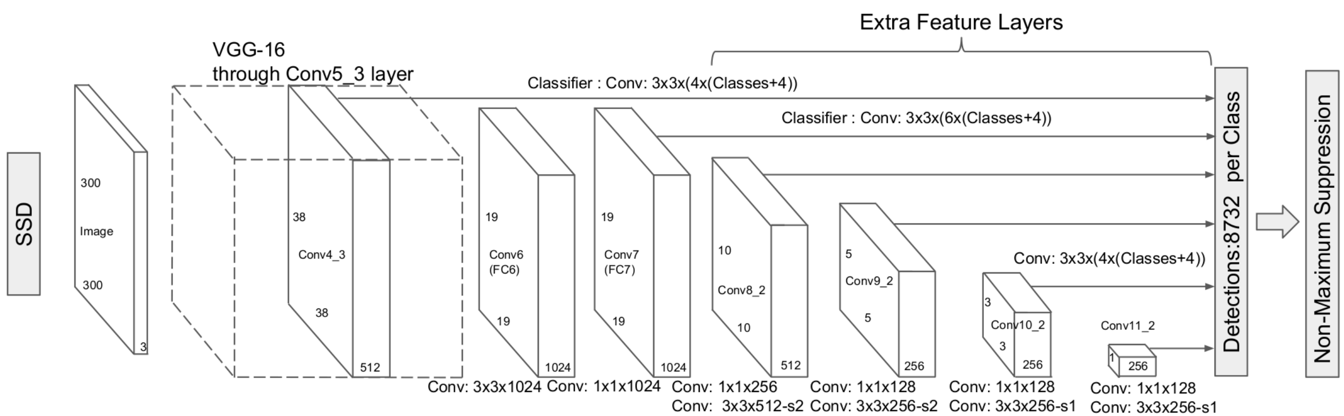


VEDAI

UAVDT

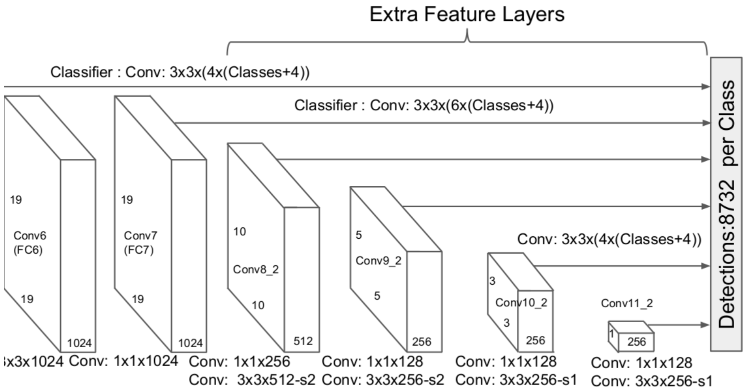
Meta-architectures

- Candidates: Faster R-CNN[1], SSD[2] and YOLOv2[3]
- SSD offers best trade-off, YOLOv2 competitive
- Quick SSD recap:
  - Base network (backbone), initially VGG-16
  - Extra feature layers
  - Convolutional layers for classification and box regression
  - Non-maximum suppression



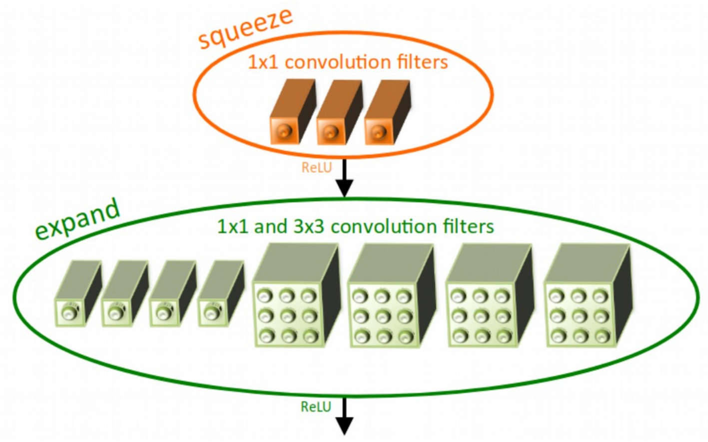
Meta-Architecture Modifications

- SSD makes use of multiple scales
  - Constant GSD
  - Use 8× downsampled feature maps
  - Found by ablation study
- Box sizes and ratios
  - Boxes have to fit the object size
  - Clustering approach similar to DSSD[4]
  - Saves filters in the prediction layers with next to no change in accuracy



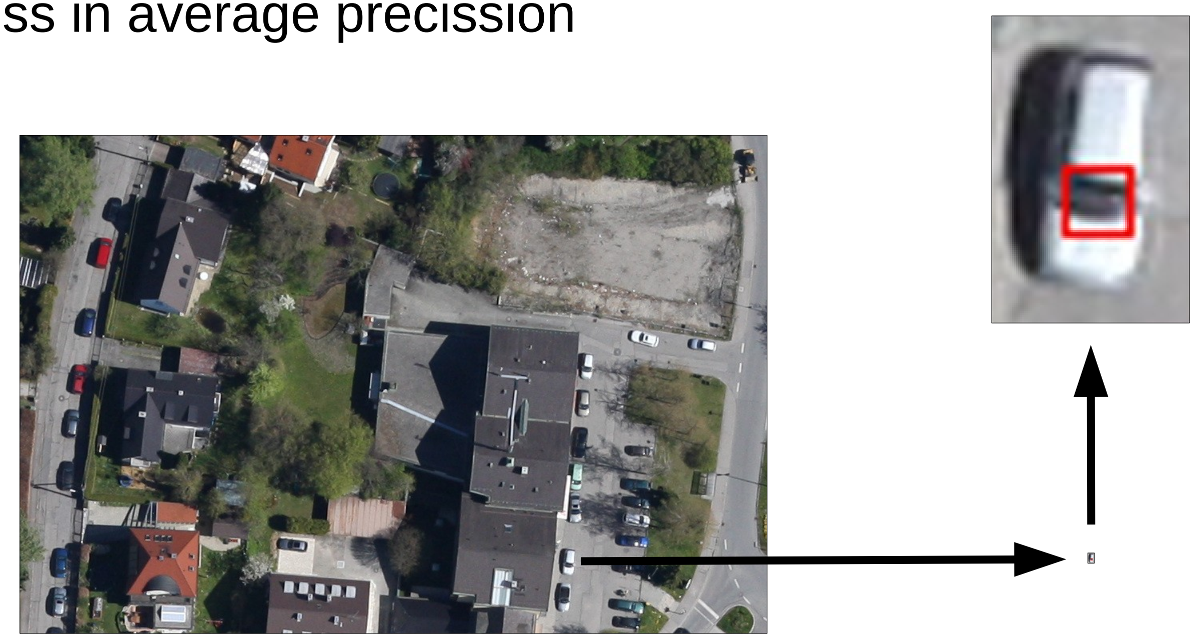
Base Networks

- Many different networks in literature
  - MobileNet[5]
  - ShuffleNet[6]
  - SqueezeNet[7]
  - ZynqNet[8]
- ZynqNet (SqueezeNet-like architecture)
  - Only standard layers
  - Strided convolution instead of pooling
  - Alternating 3×3 and 1×1 for squeeze layer
- Additional changes
  - No ReLU after squeeze layer
  - ELU instead of ReLUs



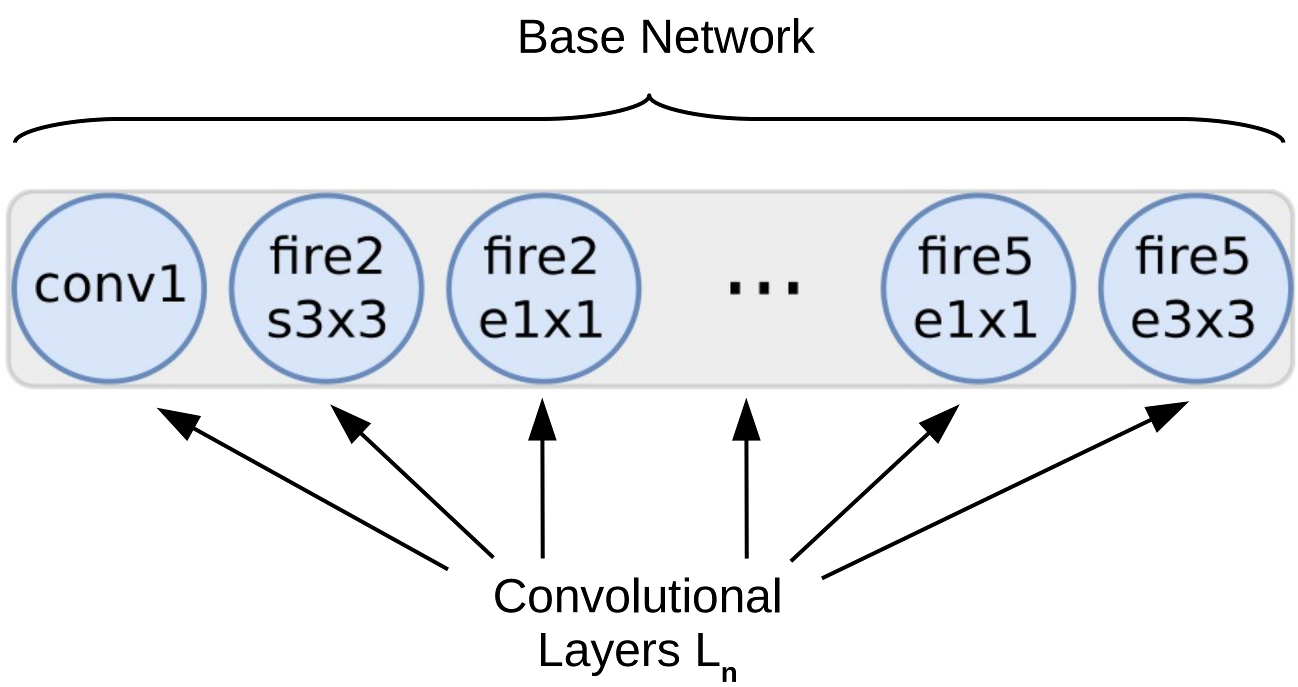
Regression and Classification Layers

- SSD uses 3×3 convolutions by default
- But strongest feature for vehicles is the windshield
- On a 8× downsampled feature map only 1 „pixel“ is covered
- 1×1 convolutions are sufficient
- No loss in average precission



Auto-Pruning

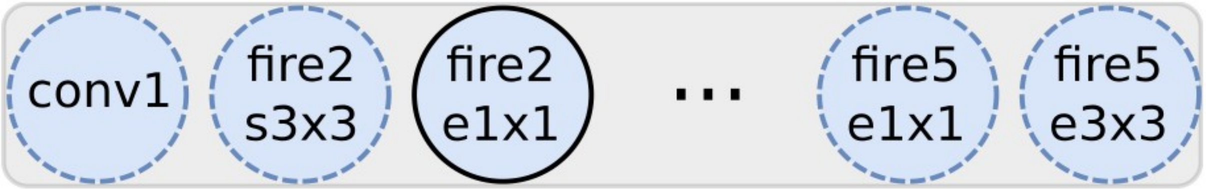
- Objective is to reduce number of filters in the base network from N to φ×N



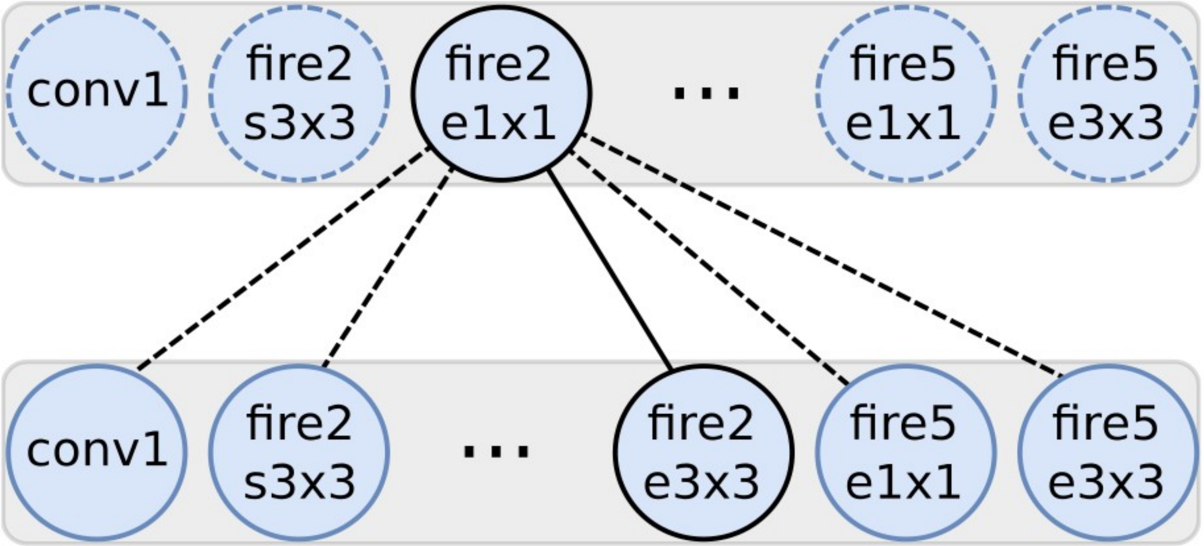
Auto-Pruning

- Iterate from back to front (later layers have more filters)
- Delete **k** filters in layer $\mathbf{L_n}$, according to l_1 norm
- Calculate validation sensitivity **S**
- Yielding tuples of (S, L_n, k)
- Remove worst tuple from network according to metric
- No retraining required!

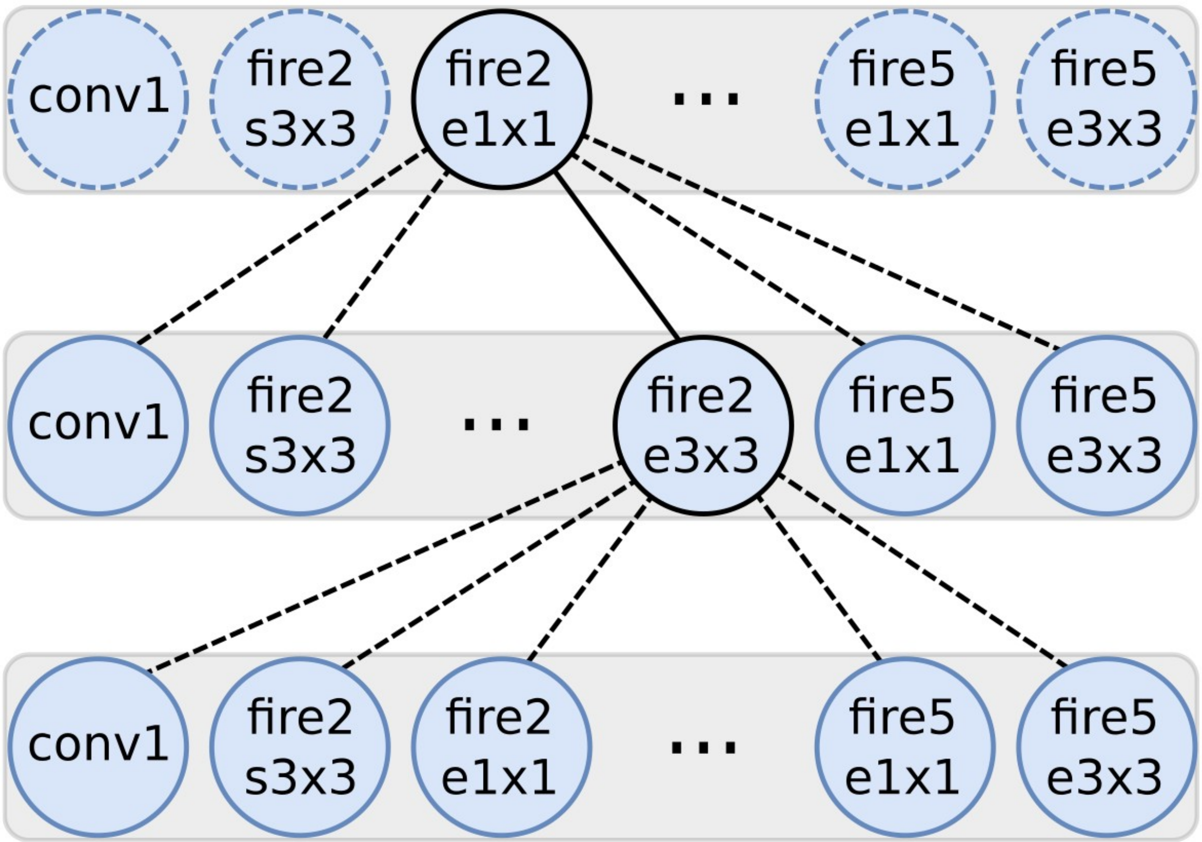
Auto-Pruning



Auto-Pruning



Auto-Pruning



Auto-Pruning

- Target is predefined φ value
  - Only one hyperparameter
  - Arbitrarily chosen: 0.5, 0.25, 0.15 etc.
  - Can be adjusted for use case
- Pruning decision can be any metric, also depending on application area (speed vs. accuracy)
- Final network only has to be finetuned once for a few iterations

Quantitative Results

- DLR 3K results:

Network	AP (%)	Inference Speed (FPS)		
		Titan X	GTX 1060	Jetson TX2
VGG	**97.2**	27.9	11.7	1.3
ZynqNet	**97.2**	184.9	81.9	14.7
UAV-Net$_{\varphi\,=\,1.000}$	**97.2**	194.1	83.8	15.9
UAV-Net$_{\varphi\,=\,0.750}$	**97.2**	225.8	98.8	18.8
UAV-Net$_{\varphi\,=\,0.500}$	97.1	265.2	116.2	22.7
UAV-Net$_{\varphi\,=\,0.250}$	95.4	342.6	153.3	31.3
UAV-Net$_{\varphi\,=\,0.150}$	91.3	410.0	181.2	38.2
UAV-Net$_{\varphi\,=\,0.075}$	11.1	**426.8**	**203.5**	**43.1**

Quantitative Results

Karlsruhe Institute of Technology

- DLR 3K results:

Network	AP (%)	Inference Speed (FPS) Titan X	GTX 1060	Jetson TX2
VGG	**97.2**	27.9	11.7	1.3
ZynqNet	**97.2**	184.9	81.9	14.7
UAV-Net$_{\varphi=1.000}$	**97.2**	194.1	83.8	15.9
UAV-Net$_{\varphi=0.750}$	**97.2**	225.8	98.8	18.8
UAV-Net$_{\varphi=0.500}$	97.1	265.2	116.2	22.7
UAV-Net$_{\varphi=0.250}$	95.4	342.6	153.3	31.3
UAV-Net$_{\varphi=0.150}$	91.3	410.0	181.2	38.2
UAV-Net$_{\varphi=0.075}$	11.1	**426.8**	**203.5**	**43.1**

Network architecture	Model Size	Parameter Count	Relative Size
VGG, 2 box sizes	30.19 MiB	7,912,316	100.0%
ZynqNet	0.89 MiB	230,782	2.9%
UAV-Net$_{\varphi=0.50}$	0.39 MiB	101,934	1.3%
UAV-Net$_{\varphi=0.15}$	**0.07 MiB**	**17,146**	**0.2%**

Quantitative Results

- VEDAI-1024 results:

Dataset	Model	AP (%)	Inference Speed (FPS) Titan X	GTX 1060	Jetson TX2
VEDAI	VGG	**96.4**	16.7	5.8	0.7
VEDAI	UAV-Net$_{\varphi=1.00}$	95.7	123.5	50.2	9.9
VEDAI	UAV-Net$_{\varphi=0.50}$	95.2	168.0	73.8	13.9
VEDAI	UAV-Net$_{\varphi=0.15}$	93.5	**256.4**	**125.9**	**22.9**

What about other setups?

- Some modifications specific to dataset: constant GSD, constant object sizes etc.
- Evaluation also shown for UAVDT
  - 1x1 regression filters not large enough
  - Single box size not sufficient anymore
  - Other modifications still useful

DLR 3K Munich

UAVDT

Quantitative Results

- UAVDT results:

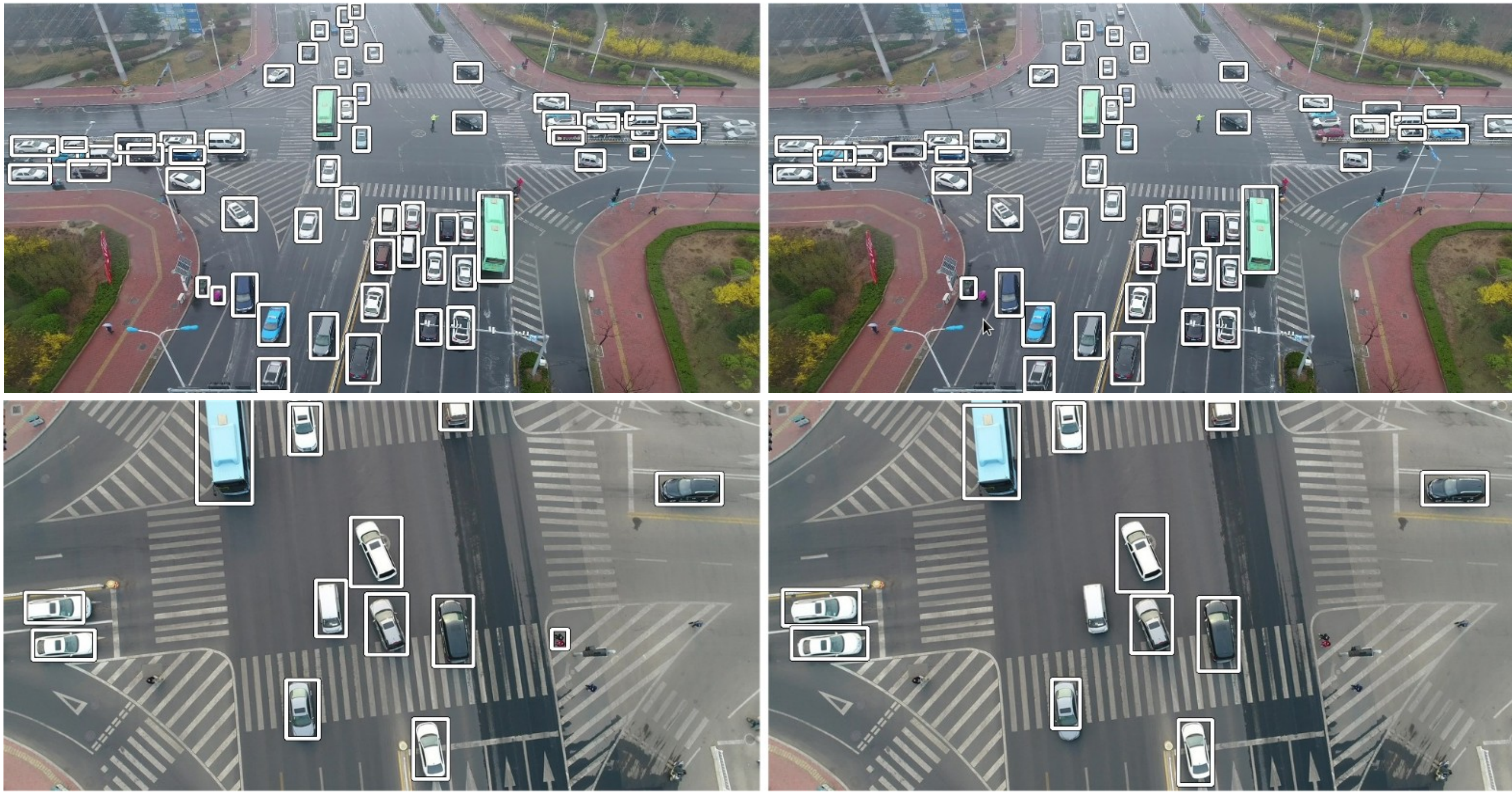
Dataset	Model	AP (%)	Inference Speed (FPS) Titan X	GTX 1060	Jetson TX2
UAVDT	R-FCN [17]	34.35	4.7	–	–
UAVDT	SSD [17]	33.62	41.6	–	–
UAVDT	Faster R-CNN [17]	22.32	2.8	–	–
UAVDT	RON [17]	21.59	11.1	–	–
UAVDT	UAV-Net$^{1\times1,c=1}_{\varphi=1.00}$	26.21	**214.0**	**98.8**	**18.3**
UAVDT	UAV-Net$^{5\times5,c=5}_{\varphi=1.00}$	**34.52**	80.1	34.7	6.6
UAVDT	UAV-Net$^{3\times3,c=4}_{\varphi=1.00}$	32.76	112.2	51.5	9.0
UAVDT	UAV-Net$^{3\times3,c=4}_{\varphi=0.50}$	31.82	132.5	69.2	11.4

Qualitative Results – DLR 3K

UAV-Net$_{\varphi=0.50}$

UAV-Net$_{\varphi=0.15}$

Qualitative Results – UAVDT



UAV-Net$_{\varphi=1.00}$ (5×5,c=5)

UAV-Net$_{\varphi=0.50}$ (3×3,c=4)

UAV-Net Summary

- Fast but still accurate vehicle detector
  - 38 FPS on DLR 3K with >90% mAP (test set)
- Ultra-low footprint for both model size and memory usage
  - As low as 0.07 MiB for DLR 3K and VEDAI
- Power envelope of <15W

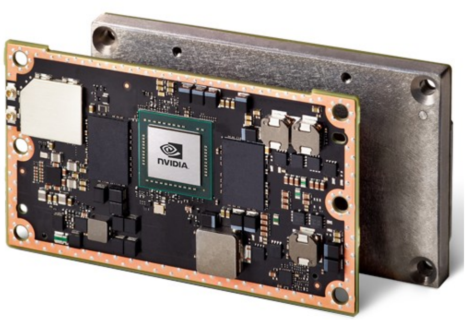


Image Sources

- Title Slide: "Artificial Intelligence & AI & Machine Learning" by mikemacmarketing is licensed under CC BY 2.0
- Drone: https://fortunedotcom.files.wordpress.com/2015/12/drone.jpg
- NVIDIA Titan V: https://www.nvidia.com/content/dam/en-zz/es_em/Solutions/geforce/TITAN/TITANV/nvidia-titan-xp-shop-625-ud.jpg
- Jetson Platform: https://devblogs.nvidia.com/wp-content/uploads/2017/03/Figure1_TX2-e1488772330657.png

  https://devblogs.nvidia.com/wp-content/uploads/2017/03/JTX2_Devkit-e1488775199359-624x615.png

References

[1] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: To-wards real-time object detection with region proposal net-works. In NIPS, 2015

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, “SSD: single shot multibox detector,” CoRR, vol. abs/1512.02325, 2015.

[3] J. Redmon and A. Farhadi.YOLO9000: better, faster, stronger. CVPR, 2017.

[4] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg.DSSD: Deconvolutional single shot detector.arXiv preprintarXiv:1701.06659, 2017

[5] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” CoRR, vol. abs/1704.04861, 2017.

[6] X. Zhang, X. Zhou, M. Lin, and J. Sun, “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” CoRR, vol. abs/1707.01083, 2017.

[7] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, “Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size,” CoRR, vol. abs/1602.07360, 2016.

[8] D. Gschwend, “Zynqnet: An fpga-accelerated embedded convolutional neural network,”, Swiss Federal Institute of Technology Zurich (ETH-Zurich), 2016.

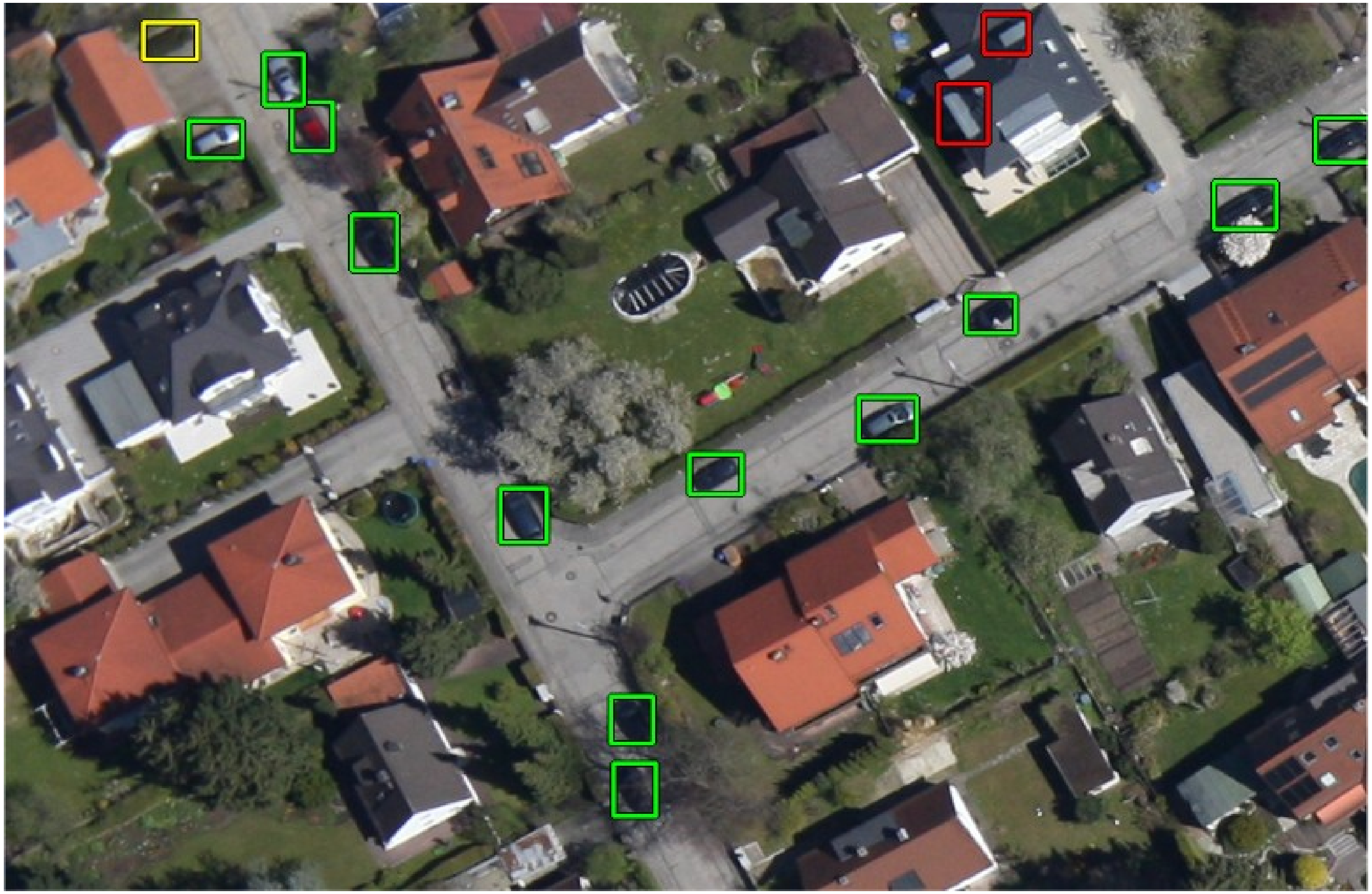
Additional Slides

Qualitative Results – DLR 3K

UAV-Net$_{\varphi=0.50}$

UAV-Net$_{\varphi=0.15}$

Qualitative Results – DLR 3K Error Cases



UAV-Net$_{\varphi=0.50}$

Qualitative Results – UAVDT

UAV-Net$_{\varphi=1.00}$ (5×5,c=5) UAV-Net$_{\varphi=0.50}$ (3×3,c=4)