TECHNISCHE UNIVERSITÄT WIEN

Institut für Computertechnik
Institute of Computer Technology

Report

by

Matthias Wess and Axel Jantsch

DNN Implementations Data Set

February 5, 2019, Vienna, Austria

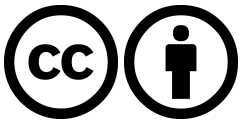
If you find this work useful, please cite it using the following BibTeX entry:

```
@TechReport{wess2019,
  author =       {Matthias Wess and Axel Jantsch},
  title =        {DNN Implementations Data Set},
  institution =  {TU Wien},
  year =         {2019},
  month =        {January},
  address =      {Gusshausstrasse 27--29 / 384, 1040 Wien, Austria}
}
```

Contact us:

matthias.wess@student.tuwien.ac.at

axel.jantsch@tuwien.ac.at



DNN Implementations and their Characteristics

Figure 1 shows power, latency, throughput, accuracy and energy efficiency characteristics of Deep Neural Network (DNN) based object detection algorithms implemented on different platforms, namely Field Programmable Gate Arrays (FPGAs), Graphic Processing Units (GPUs) and mobile Graphic Processing Units (mGPUs). All implementations have been evaluated with the ImageNet dataset. The figures are either from publications in the period 2016-2018 or from Nvidia and Xilinx websites accessed in 2018, as listed in the references below.

Table 1 allows to identify the references for each data point of the plots in figure 1, as the entries in the table constitute the coordinates in the plots. For instance, the first table entry, Jiao et al. 2017, with a power consumption of 2.3W and a latency of 9.41ms identifies in the top left latency-power plot of figure 1 as the leftmost +, right on the Pareto curve.

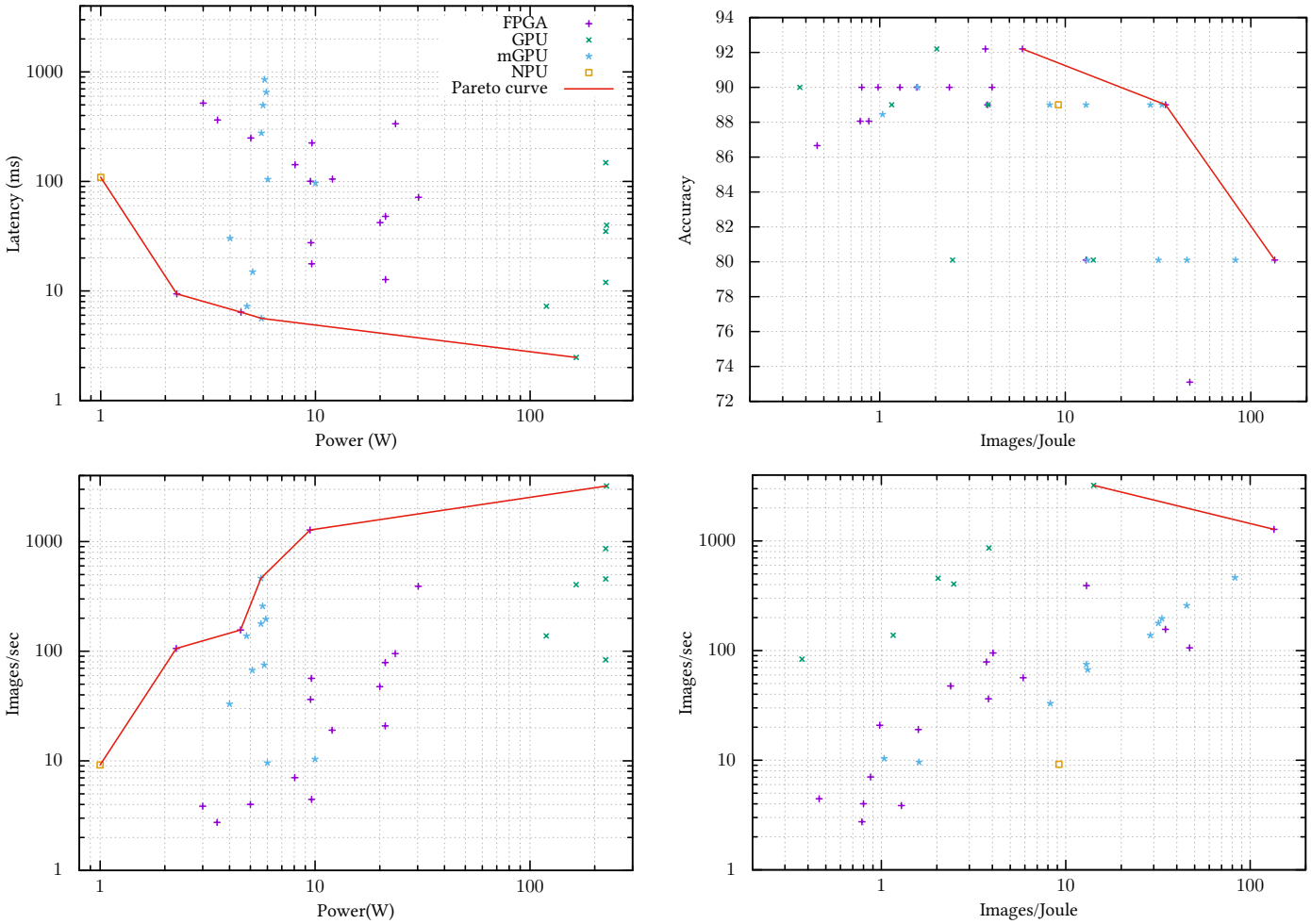


Figure 1: DNN implementations evaluated with the ImageNet dataset as reported in recent publications from 2016-2018 and the Xilinx and Nvidia websites accessed in 2018.

It is remarkable that the implementations vary 2 orders of magnitude in power consumption, 3 orders in throughput as measured in images per second, over 2 orders of magnitude in latency per image, and over 2 orders of magnitude in energy efficiency. Moreover, the achieved accuracy for the top 5 predictions is between 73% and 92%.

GPU based implementations have the highest power consumption and also the highest throughput, even though some GPU based solutions exhibit high power consumption but only moderate performance. Both GPU and mGPU based

implementations have relatively narrow range in the power consumption but vary quite a bit with respect to the other metrics. In contrast, FPGA based solutions stretch over larger regions in all metrics considered in Figure 1.

The figures display a Pareto curve that connects the Pareto optimal points. Most of the Pareto optimal solutions are FPGA based (64%), indicating their flexibility facilitating customization for particular objectives and applications, by using arbitrary bitwidth and architectures optimized for specific network structures.

Accuracy Plots

The plots in figure 2 set the achieved accuracy in relation to power consumption, latency and throughput. None of these parameters are strongly correlated to accuracy which seem to suggest, that the solutions of our dataset ave not been subject to a trade-off among those parameters. Higher accuracy has not been achieved at the expense of higher latency, lower throughput or higher power. However, the drawn Pareto curve still seems to suggest that such a trade-off may exist, but the solutions in our dataset simply have not explored this trade-off. consumption.

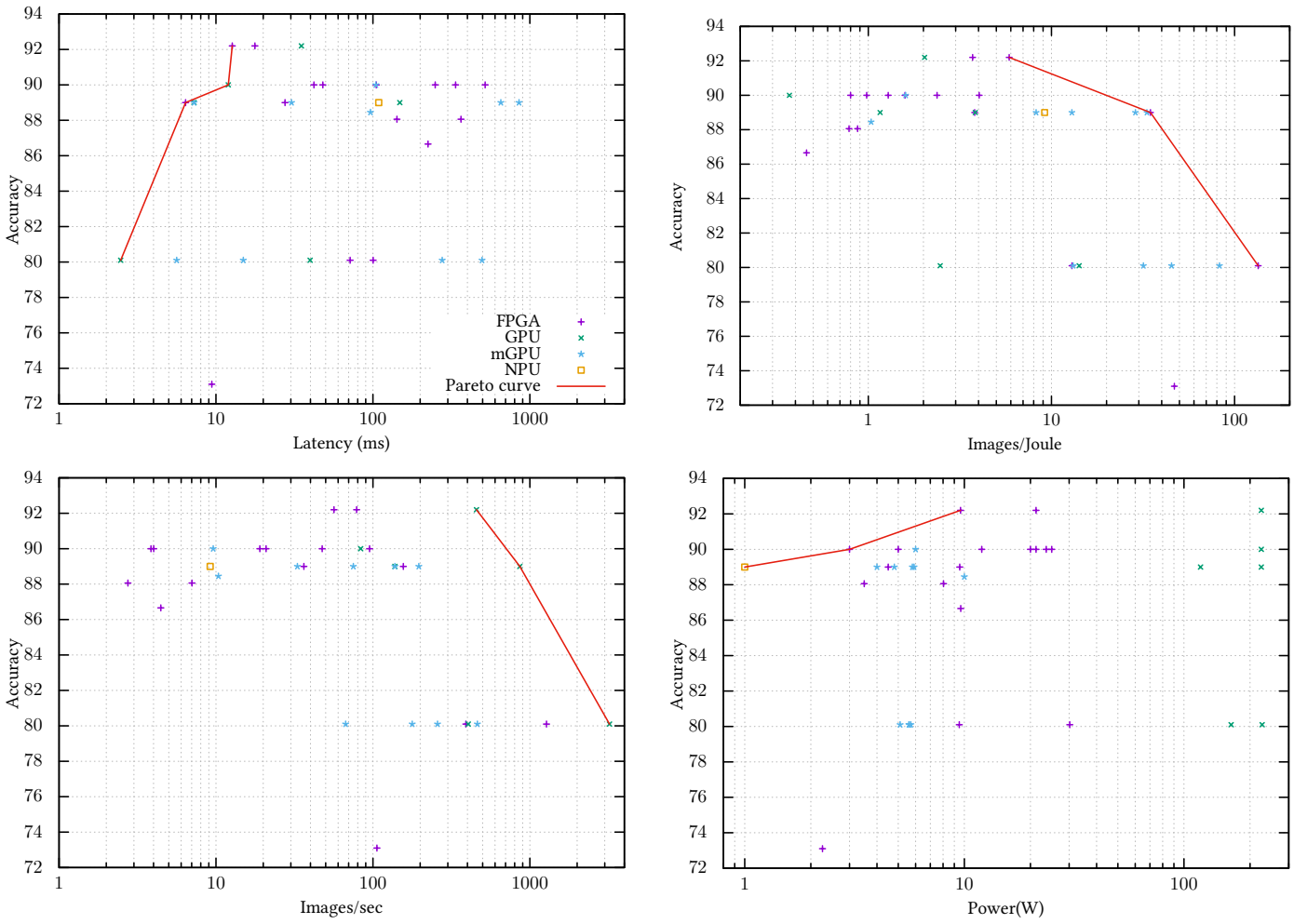


Figure 2: Plots relating accuracy to power, energy efficiency, latency and throughput.

Power Plots

The plots in figure 3 set the power consumption in relation to accuracy, latency, energy efficiency and throughput.

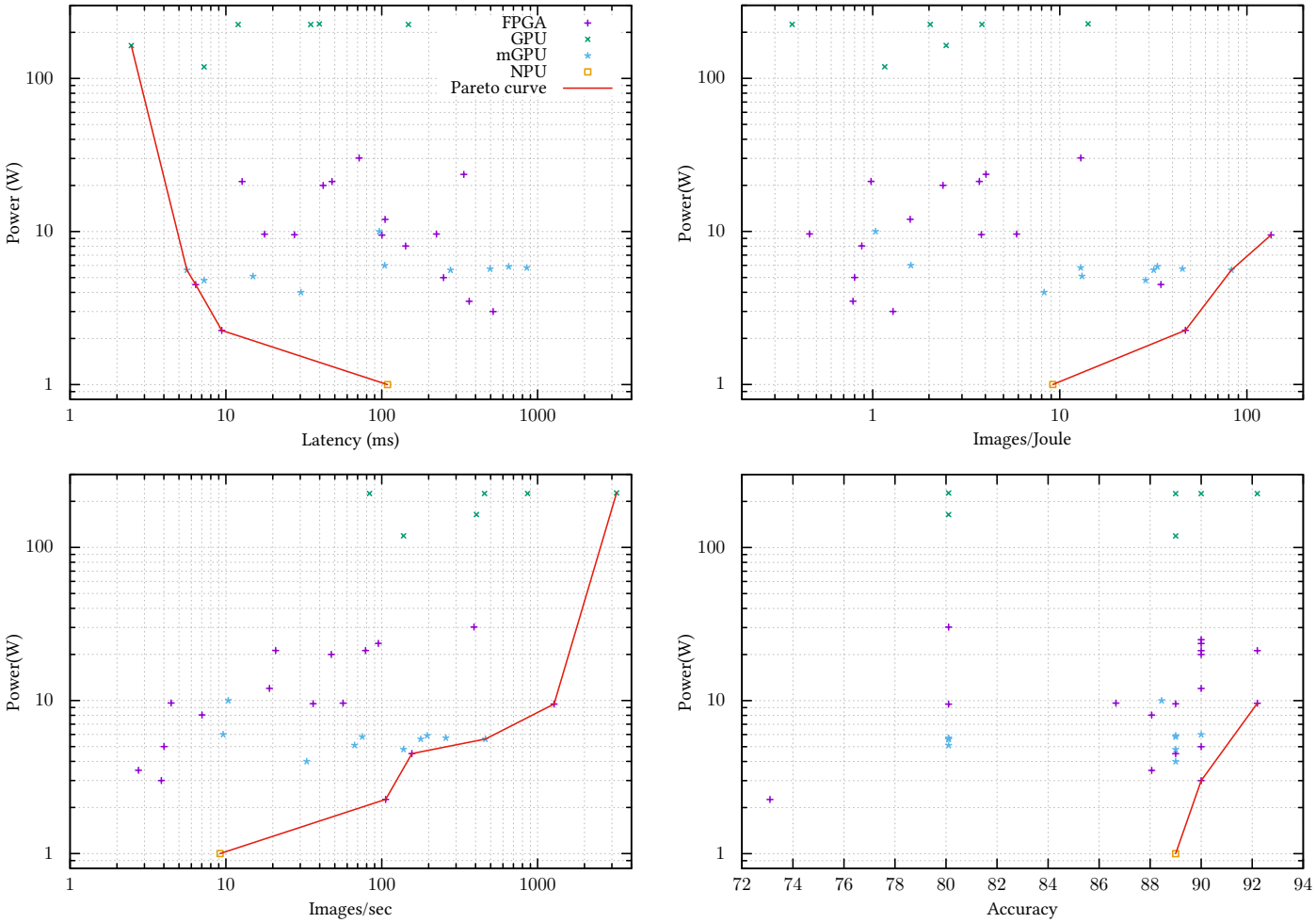


Figure 3: Plots relating power consumption to energy efficiency, latency, throughput, and accuracy.

Energy Efficiency Plots

The plots in figure 4 set the energy efficiency (images/Joule) in relation to accuracy, latency, power consumption and throughput.

The correlation in the throughput/energy efficiency plot as suggested by the plot at the lower left of figure 4, is a bit unexpected. The higher the throughput the higher the energy efficiency. However, there is no correlation with latency.

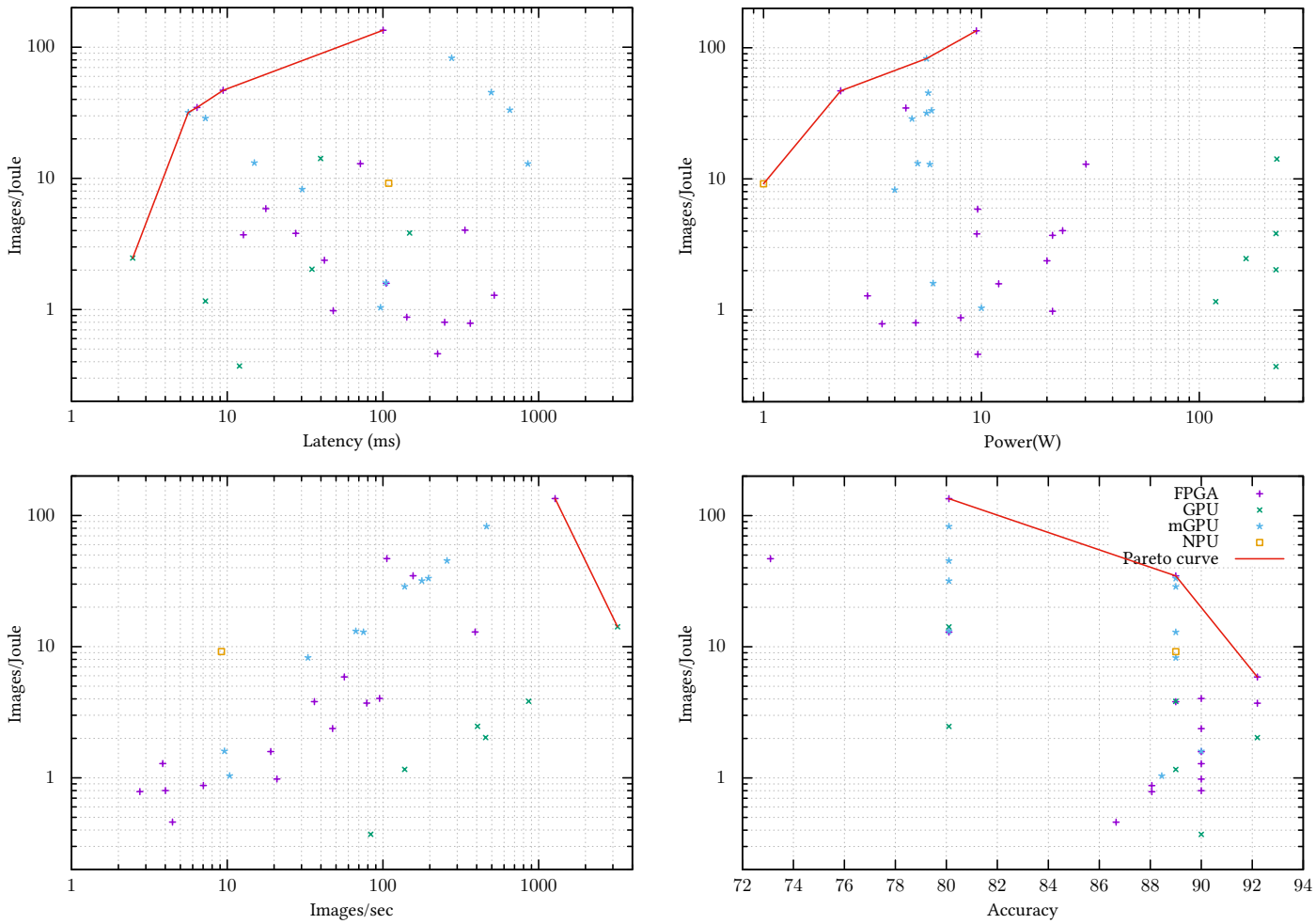


Figure 4: Plots relating energy efficiency, measured in images/Joule, to power consumption, latency, throughput, and accuracy.

Throughput Plots

The plots in figure 5 set the throughput in relation to accuracy, latency, energy efficiency and power consumption.

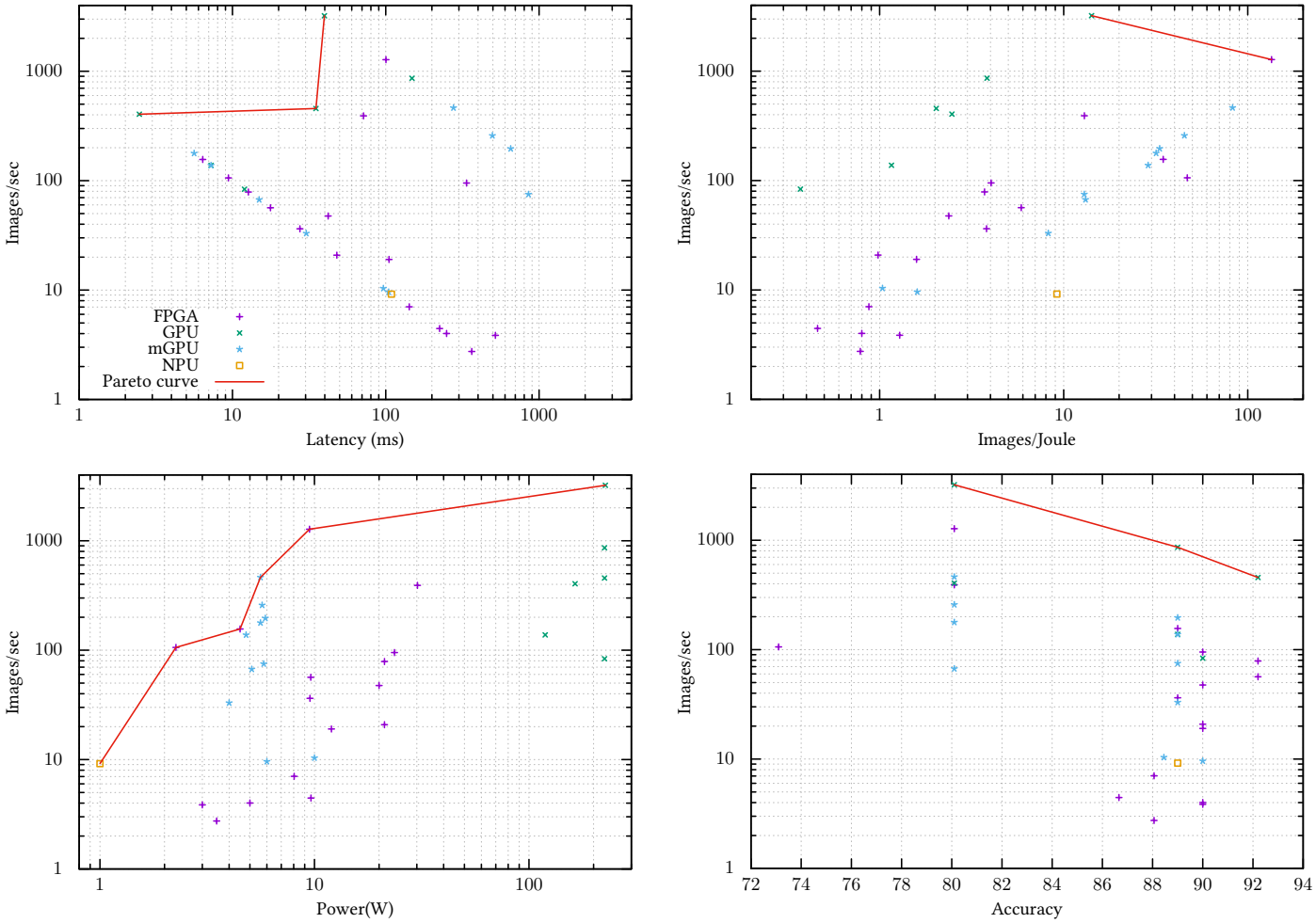


Figure 5: Plots relating throughput, measured in images/sec, to power consumption, energy efficiency, latency, and accuracy.

Latency Plots

The plots in figure 6 set the latency in relation to power consumption, accuracy, energy efficiency and throughput.

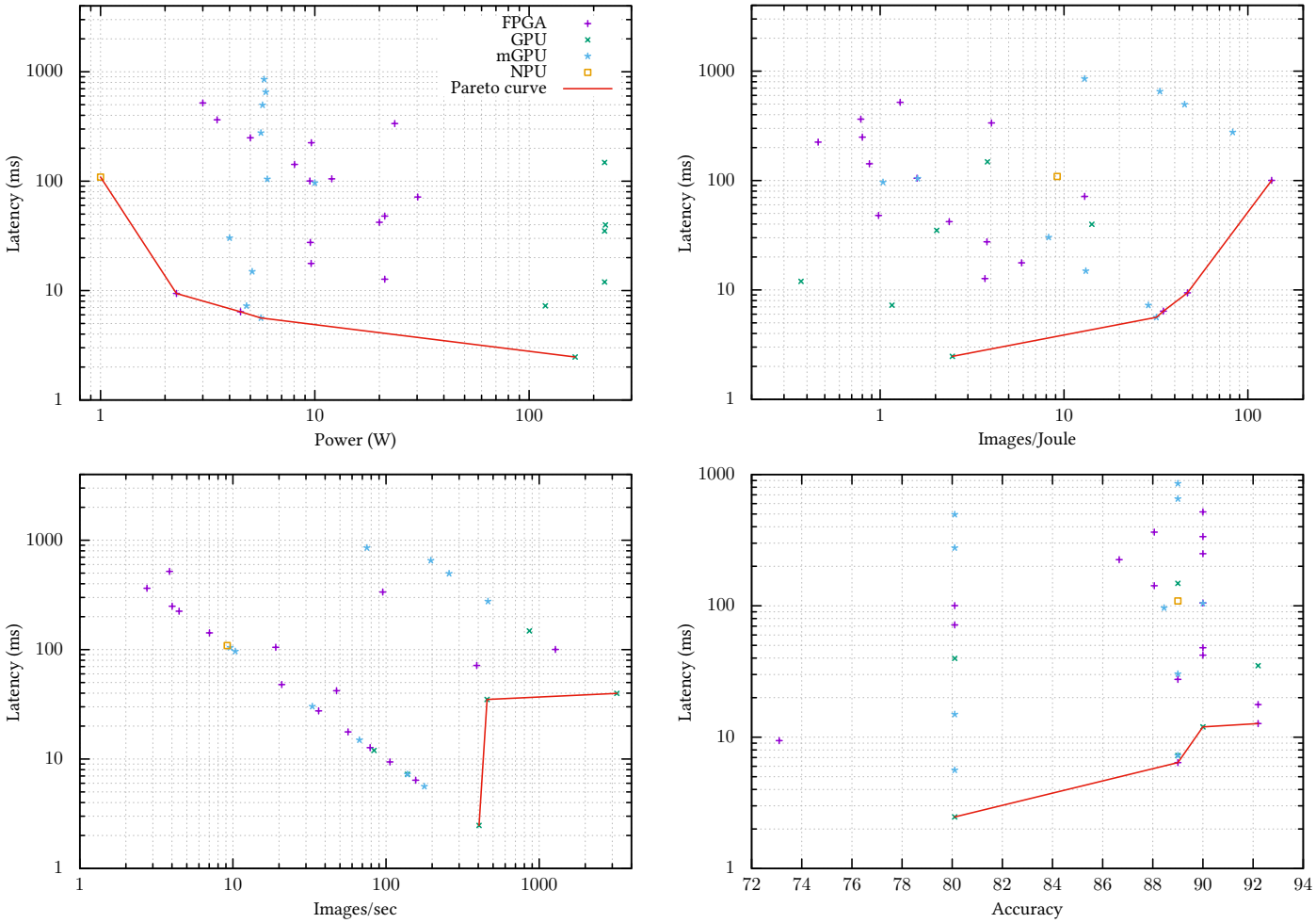


Figure 6: Plots relating latency to energy efficiency, power consumption, throughput, and accuracy.

Table of References

Table 1: Database

Title	Network	Type	Power (W)	Latency (ms)	Img./sec	Img./J	TOP5 (%)
Jiao et al. 2017	DoReFa-Net	FPGA	2.3	9.41	106.0	46.90	73.1
Lu et al. 2017	VGG-16	FPGA	23.6	336.33	95.1	4.03	90.0
Qiu et al. 2016	VGG-16	FPGA	9.6	224.60	4.5	0.46	86.7
H. Li et al. 2016	AlexNet	FPGA	30.2	71.61	391.0	12.95	80.1
Ma et al. 2018	VGG-16	FPGA	21.2	47.97	20.8	0.98	90.0
Ma et al. 2018	ResNet-50	FPGA	21.2	12.70	78.7	3.71	92.2
Podili, Zhang, and Prasanna 2017	VGG-16	FPGA	8.0	142.30	7.0	0.87	88.1
Yu et al. 2017	VGG-16	FPGA	20.0	42.11	47.5	2.37	90.0
Guo, Sui, Qiu, Yu, et al. 2018	VGG-16	FPGA	3.5	364.00	2.7	0.78	88.1
Venieris and Bouganis 2016	VGG-16	FPGA	5.0	249.38	4.0	0.80	90.0
Gokhale et al. 2017	GoogLeNet	FPGA	9.5	27.55	36.3	3.81	89.0
Gokhale et al. 2017	ResNet-50	FPGA	9.6	17.70	56.5	5.88	92.2
Gokhale et al. 2017	AlexNet	FPGA	9.5	100.30	1276.2	134.62	80.1
Guo, Sui, Qiu, Yao, et al. 2018	VGG-16	FPGA	3.0	519.00	3.9	1.28	90.0
Guo, Sui, Qiu, Yao, et al. 2018	VGG-16	FPGA	12.0	105.06	19.0	1.59	90.0
Kathail 2017	GoogLeNet	FPGA	4.5	6.40	156.3	34.72	89.0
Guo, Sui, Qiu, Yu, et al. 2018	VGG-16	mGPU	10.0	96.50	10.4	1.04	88.5
Behrends et al. 2015	AlexNet	mGPU	5.1	14.93	67.0	13.14	80.1
Behrends et al. 2015	AlexNet	mGPU	5.7	496.12	258.0	45.26	80.1
Franklin 2017	AlexNet	mGPU	5.6	5.62	178.0	31.79	80.1
Franklin 2017	AlexNet	mGPU	5.6	276.46	463.0	82.68	80.1
Behrends et al. 2015	GoogLeNet	mGPU	4.0	30.30	33.0	8.25	89.0
Behrends et al. 2015	GoogLeNet	mGPU	5.8	853.33	75.0	12.93	89.0
Franklin 2017	GoogLeNet	mGPU	4.8	7.25	138.0	28.75	89.0
Franklin 2017	GoogLeNet	mGPU	5.9	653.06	196.0	33.22	89.0
charlyng 2018	VGG-16	mGPU	6.0	104.40	9.6	1.60	90.0
Behrends et al. 2015	AlexNet	GPU	164.0	2.47	405.0	2.47	80.1
Behrends et al. 2015	AlexNet	GPU	227.0	39.80	3216.0	14.17	80.1
charlyng 2018	VGG-16	GPU	225.0	11.98	83.5	0.37	90.0
Behrends et al. 2015	GoogLeNet	GPU	119.0	7.25	138.0	1.16	89.0
Behrends et al. 2015	GoogLeNet	GPU	225.0	148.32	863.0	3.84	89.0
Johnson 2018	ResNet-50	GPU	225.0	35.03	456.8	2.03	92.2
Rachakonda 2018	GoogLeNet	NPU	1.0	108.95	9.2	9.18	89.0

Bibliography

[Beh+15] R. Behrends, L. K. Dillon, S. D. Fleming, and R. E. K. Stirewalt. *GPU-Based Deep Learning Inference: A Performance and Power Analysis.* 2015. URL: https://www.nvidia.com/content/tegra/embedded-systems/pdf/jetson_tx1_whitepaper.pdf (visited on 12/18/2018).

[cha18] charlyng. *Embedded-Deep-Learning.* 2018. URL: https://github.com/charlyng/Embedded-Deep-Learning/tree/master/Benchmark-Performance (visited on 12/18/2018).

[Fra17] D. Franklin. *NVIDIA Jetson TX2 Delivers Twice the Intelligence to the Edge.* 2017. URL: https://devblogs.nvidia.com/jetson-tx2-delivers-twice-intelligence-edge/ (visited on 12/18/2018).

[Gok+17] V. Gokhale, A. Zaidy, A. X. M. Chang, and E. Culurciello. "Snowflake: A Model Agnostic Accelerator for Deep Convolutional Neural Networks". In: *arXiv preprint arXiv:1708.02579* (2017).

[Guo+18a] K. Guo, L. Sui, J. Qiu, S. Yao, S. Han, and Y. Wang. *From Model to FPGA: Software-Hardware Co-Design for Efficient Neural Network Acceleration.* 2018. URL: https://www.hotchips.org/wp-content/uploads/hc_archives/hc28/HC28.22-Monday-Epub/HC28.22.40-Vision-Image-Epub/HC28.22.411-Neural-Net-Accleration-Yao-DeePhi-0821.pdf (visited on 12/18/2018).

[Guo+18b] K. Guo, L. Sui, J. Qiu, J. Yu, J. Wang, S. Yao, S. Han, Y. Wang, and H. Yang. "Angel-Eye: A Complete Design Flow for Mapping CNN Onto Embedded FPGA". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37.1 (2018), pp. 35–47.

[Jia+17] L. Jiao, C. Luo, W. Cao, X. Zhou, and L. Wang. "Accelerating Low Bit-Width Convolutional Neural Networks with Embedded FPGA". In: *Field Programmable Logic and Applications (FPL), 2017 27th International Conference on.* IEEE. 2017, pp. 1–4.

[Joh18] J. Johnson. *CNN Benchmarks.* 2018. URL: https://github.com/jcjohnson/cnn-benchmarks (visited on 12/18/2018).

[Kat17] V. Kathail. *Xilinx Caffe to Zynq: State-of-the-Art Machine Learning Inference Performance in Less Than 5 Watts.* 2017. URL: https://www.embedded-vision.com/sites/default/files/webinars/May%2024,%202017%20Webinar.pdf (visited on 12/18/2018).

[Li+16] H. Li, X. Fan, L. Jiao, W. Cao, X. Zhou, and L. Wang. "A High Performance FPGA-based Accelerator for Large-Scale Convolutional Neural Networks". In: *2016 26th International Conference on Field Programmable Logic and Applications (FPL).* IEEE. 2016, pp. 1–9.

[Lu+17] L. Lu, Y. Liang, Q. Xiao, and S. Yan. "Evaluating Fast Algorithms for Convolutional Neural Networks on FPGAs". In: *2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE. IEEE, 2017, pp. 101–108.

[Ma+18] Y. Ma, Y. Cao, S. Vrudhula, and J.-s. Seo. "Optimizing the Convolution Operation to Accelerate Deep Neural Networks on FPGA". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 26.7 (2018), pp. 1354–1367.

[PZP17] A. Podili, C. Zhang, and V. Prasanna. "Fast and Efficient Implementation of Convolutional Neural Networks on FPGA". In: *IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP)*. IEEE. IEEE, 2017, pp. 11–18.

[Qiu+16] J. Qiu, J. Wang, S. Yao, K. Guo, B. Li, E. Zhou, J. Yu, T. Tang, N. Xu, S. Song, et al. "Going Deeper with Embedded FPGA Platform for Convolutional Neural Network". In: *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM. ACM Press, 2016, pp. 26–35.

[Rac18] R. Rachakonda. *Intel Movidius Inference performance*. 2018. URL: https://ncsforum.movidius.com/discussion/149/inference-performance (visited on 12/18/2018).

[VB16] S. I. Venieris and C. Bouganis. "fpgaConvNet: A Framework for Mapping Convolutional Neural Networks on FPGAs". In: *IEEE Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, May 2016, pp. 40–47.

[Yu+17] J. Yu, Y. Hu, X. Ning, J. Qiu, K. Guo, Y. Wang, and H. Yang. "Instruction driven cross-layer CNN accelerator with winograd transformation on FPGA". In: *International Conference on Field Programmable Technology (ICFPT)*. IEEE. 2017, pp. 227–230.