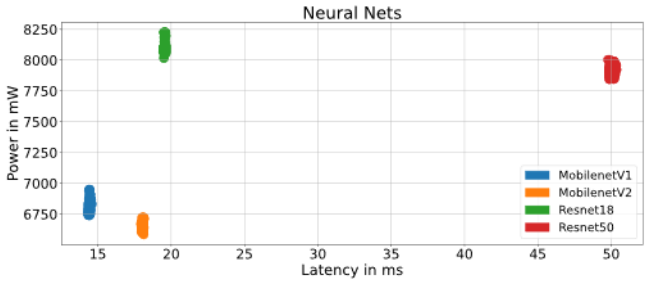

Figure 8. Power over latency of different Nets

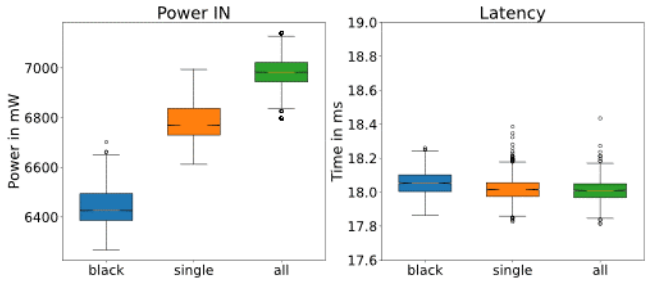

Figure 9. Image type effect on power consumption

In Fig. 8, the power per inference over the latency is displayed. Tab. II shows the accuracies of the networks. When comparing the MobileNets, we notice a slightly higher consumption in MobileNetV1. Furthermore, MobileNetV1 is 4ms faster than MobileNetV2. ResNet50 is the slowest, with a latency of 50ms. It is nearly 3 times slower than ResNet18. The power consumption of the two ResNets is similar. ResNet18 consumes a little bit more power than ResNet50.

In general, MobileNets are faster and more energy-efficient than ResNets at a comparable accuracy. The fastest inference and most energy-efficient classification can be achieved with MobileNetV1. When power is optimized, MobileNetV2 is 1% better than V1. Smaller ResNets seem to be outdated; not only do they need much more power, but also, the latency is higher compared to MobileNets.

TABLE II. OPTIMIZED SETTINGS

	MobileNet V1	**MobileNet V2**	**ResNet18**	**ResNet50**
Top-1 Accuracy	70.6%	72.0%	72.12%	77.15%

How power changes with different inputs can be seen in Fig. 9. We chose three test settings: a black image, a single random image from ImageNet, and 100 images from ImageNet. We collect power values from over 2000 inference cycles for every test case. After each inference cycle, we load a new or the same image into memory. We execute each of the 100 images 20 times. We execute the black and the single image 2000 times. While latency does not differ significantly, the input power increases 10% for a randomly picked image compared to only a black image. A fixed single random image increases power consumption by 5%. The cause for this might be that the black image results in simple operations with 0, which results in less flipping of transistors.

Then, we want to find out whether the power consumption of a layer changes when it is extracted from a network. A neural network is composed of individual layers. A question is whether a particular layer consumes the same amount of power in a single-layer network as in the original network. To answer that, we compose a small network that consists of the first layers of MobileNetV2 as a single layer network. To get comparable values, we analyze collected layer information from the parsed ONNX network. We look at every layer to see the input and output size as well as the layer type.

However, as TensorRT limits reverse engineering, we cannot get in-depth information like the kernel size of a convolution layer or type of activation. It is also not possible to delete layers to reduce the network to one layer of interest. Therefore, we extracted the missing layer information from TensorRTx Github page. Unlike the TensorRTx rebuild of MobileNetV2, RELU6, which is not supported by TensorRT, was supplemented with a standard RELU function.

In Fig. 10, we show the power and latency of the small network. To the left, the boxes contain mapped values of the layers. To the right, the boxes contain values from the single-layer networks. When comparing, we see a big power difference. Mapped values are much higher at 6W, while the single-layer values have a power consumption of about 4W to 5W. In general, the power consumption of layers embedded in a network does not match the power consumption of single-layer networks. Single-layer networks also have an initial layer, which is the input reformatter. This layer sometimes takes longer than the actual layer. Therefore, we take time values from the Nvidia profiler to get the values for the specific layer. While power values cannot be directly compared, latency values are comparable. The latency of the layers is equal in the entire network and single-layer networks. Energy values are also not the same due to the difference in power.

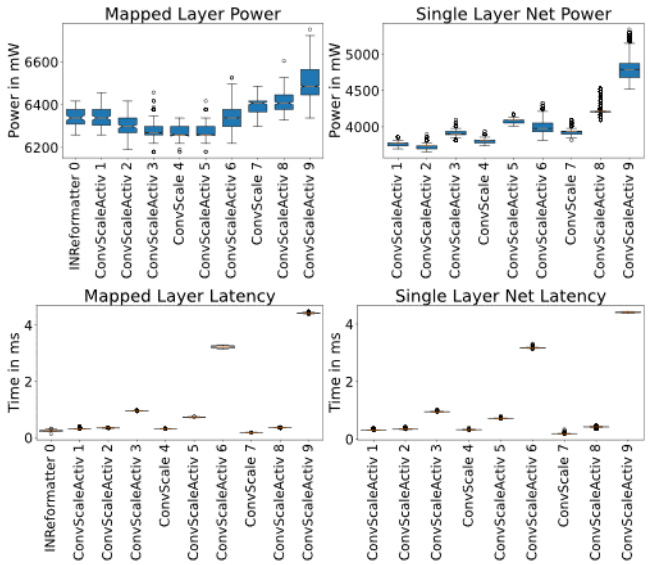

Figure 10. Power, latency and energy of single layers of MobileNetV2

A single-layer network can be used to compare latency but not to get the power consumption or the energy consumption of a layer in a network of more layers. The neighboring layers contribute significantly to power consumption.

Next, we provide an in-depth power analysis of the convolutional layer. We built and measured single-layer networks of a convolutional layer. We look at three parameters: (1) kernel size, (2) stride, and (3) output shape. The basic settings are kernel and stride to 1x1 and output and input shape to 3x224x224. The range in which kernel size and stride change are from 1x1 to 5x5. The output shape is set from 3x224x224 to 48x224x224, only change of one dimension. Padding is set so that no change in size occurs. Fig. 11 shows the effect on power consumption. Power increases with the kernel size, as already seen in the single-layer networks for MobileNet. A bigger kernel size results in more operations, therefore, more power. For stride, a decrease in power can be seen. Stride reduces the size of the output tensor; therefore, less power is needed. An increment in output shape causes an increment in power. Kernel size and the output shape have the most significant effect on power. When optimizing a network for power, a reduction of those parameters should be considered.

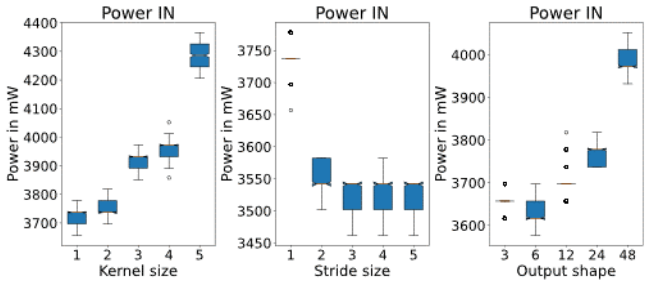

Figure 11. Power of different kernel, stride size and output shape of a convolutional network

To summarize, MobileNets are very energy efficient with comparable accuracy. Power consumption depends on the inputs: a black image needs less power than images with color. When using single-layer networks, latency is comparable with layers embedded in a network, but power differs significantly. Regarding the network structure, a smaller kernel in convolution layers, a stride setting greater than one and small output shape results in a lower power consumption

IV. Conclusion

We provided a reliable method to measure the power consumption of entire neural networks as well as single layers of a neural network on the NVIDIA Jetson Nano. The challenge was to get useful measurements with the limited time resolution of the onboard sensors, and we solved it by collecting data points from multiple inferences.

Measurements on four different hardware settings provide information on how inference can be optimized to power, latency, or energy. The settings are active CPU cores, frequency of CPU and GPU, and the effect of asynchronous mode. Optima have been found for MobileNetV2; for other networks, these optima can lie elsewhere. The results may also change when combining multiple parameters.

We tested how the network structure affects power consumption. Power differs from network to network, and it changes with different inputs. When rebuilding a layer of a network in a single-layer network, the power consumption is not the same as in the entire network. The neighboring layers do play an essential role in power consumption.

This work with accurate power measurements serves as a base for the development of power estimation models of the NVIDIA Jetson Nano. Further, we plan to extend our measurements with the same methods as in this paper to other related NVIDIA boards, such as TX2 and Xavier.

Acknowledgment

The financial support by the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology, and development are gratefully acknowledged.

References

[1] J. Kaster, J. Patrick, and H. S. Clouse. “Convolutional neural networks on small unmanned aerial systems,” in Aerospace and Electronics Conference (NAECON), 2017 IEEE National. IEEE, 2017, pp. 149–154.

[2] S. Luo, H. Lu, J. Xiao, Q. Yu, and Z. Zheng, “Robot detection and localization based on deep learning,” in Chinese Automation Congress (CAC), 2017, 2017, pp. 7091–7095.

[3] Mittal, S. A Survey on optimized implementation of deep learning models on the NVIDIA Jetson platform. Journal of Systems Architecture, 97, 2019, pp. 428-442.

[4] E. Cai, Da-C. Juan, D. Stamoulis and D. Marculescu, “NeuralPower: Predict and Deploy Energy-Efficient Convolutional Neural Networks”, conference paper at ACML 2017

[5] Ž. Nakutis, “Embedded Sysems Power Consumption Measurement Methods Overview”, Kaunas Univerity of Technology, ISSN 1392-1223 MATAVIMAI. 2009. Nr. 2(44)

[6] C. F. Rodrigues, G. Riley and M. Luján, “SyNERGY: An energy measurement and prediction framework for Convolutional Neural Networks on Jetson TX1”, Int'l Conf. Par. and Dist. Proc. Tech. and Appl. | PDPTA'18 | 375

[7] C. F. Rodrigues, G. Riley and M. Luján, “Fine-Grained Energy Profiling for Deep Convolutional Neural Networks on the Jetson TX1”, 978-1-5386-1233-0/17/$31.00 ©2017 IEEE

[8] D. Kang, D. Kang, J. Kang, S. Yoo and S. Ha, „Joint Optimization of Speed, Accuracy, and Energy for Embedded Image Recognition Systems”, 978-3-9819263-0-9/DATE18/┐ c 2018 EDAA

[9] C. Xiao, G. Chen and W. G. H. Odendaal, “Overview of Power Loss Measurement Techniques in Power Electronics Systems”, IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, VOL. 43, NO. 3, MAY/JUNE 2007