Investigation of Energy Efficient CNNs

Kaan Güler

Computer Engineering
TED University
Ankara, Türkiye
Email: kaan.guler@tedu.edu.tr

***Abstract*—In today's world, costs need to decrease in order for systems such as autonomous vehicles and robots to enter our lives more easily. One way to do this is to make environmental perception processes consume less power in the hardware.In this work, we investigate a power consumption versus MAC count and parameter size for RTX 2060 and Rtx 3070.We evaluate across 16 different convolutional neural networks (CNNs) models on Cifar-10 dataset.We find that the MAC number and parameter number have a direct impact, the GPU trained does not matter for accuracy, and the architecture of the model affects the inference time.**

I. Introduction

In many robotic applications , there is a need to achieve certain accuracy within latency. However ,in robotic devices, power consumption is additional constraint for usage in real life. To overcome this issue,balancing between power consumption, latency and accuracy is the key factor. For applications like object detection used in autonomous RC cars with limited resource , low power consuming CNNs is a meaningful choice as transformers have larger model sizes and higher memory requirements.In this work , we will look at the efficient CNNs in terms of power consumption.

The research question is to analyze how different models and metrics impact the power consumption and relation between power consumption and number of MAC and parameters. Consequently, We can find a optimal model for given tasks.

We use the following methodology: As a first step, we select different and same models but with different depths used in different tasks such as object detection and image classification as backbone.

Then, we determine metrics to compare different CNN models to analyze what effect the power consumption .Finally , We decide frameworks to measure power consumption and other metrics. From that , we can conclude which metric have the highest impact and which model give the optimal based on MAC , number of parameters and power consumption.

We contribute to research in this area in the following ways:

- We provide a measurement for different GPUs to understand importance of hardware.
- We provide insights into how CNN layers , as well as number of parameters and MAC, influence power consumption on the rtx 2060 and rtx 3070.
- We shows optimal model for the RTX 2060 and rtx 3070 of accuracy ,MAC , number parameters, latency, energy- and power consumption per inference run.

II. Related Work

Environmental perception systems are critical for autonomous vehicles and robots, and the trade-off between accuracy, latency, and power consumption has been widely studied. This section reviews related works that address efficient convolutional neural networks (CNNs), hardware optimization, and energy-aware computation.

A. Power-Efficient Deep Learning Models

Several studies have focused on designing CNN architectures optimized for power efficiency while maintaining high accuracy. Sandler et al. [1] introduced MobileNetV2, a lightweight architecture leveraging depthwise separable convolutions to reduce computational complexity and energy consumption. Similarly, Tan and Le [8] extended this concept with EfficientNet, which scales network width, depth, and resolution for better performance-power trade-offs. These studies highlight the importance of reducing MACs (Multiply-Accumulate operations) and parameter size in minimizing energy use.

B. Hardware-Specific Optimizations

The performance of CNNs is hardware-dependent, making it essential to evaluate models on target platforms. Mittal [3] explored CNN optimizations on edge devices like NVIDIA Jetson boards, emphasizing the impact of power-efficient inference for real-time applications. Zhang et al. [4] analyzed hardware-specific constraints and proposed quantization techniques to reduce memory usage and computation overhead on GPUs and edge devices, such as the Jetson Nano and Orin.

C. Summary

While significant progress has been made in optimizing CNNs for power efficiency, there remains a need for a comprehensive analysis of modern architectures across multiple hardware platforms. This work builds upon the findings of prior studies by evaluating 16 CNN models on RTX 2060, and RTX 3070, providing new insights into the interplay between MACs, parameters, and power consumption.

III. Approach

This study employs a systematic methodology to evaluate the power consumption , latency ,and accuracy performance of various Convolutional neural network models across different hardware platforms. The main focus is understanding the relationship between CNN architectures and factors such as power consumption , MAC count , and parameter size . The following steps outline our approach in detail:

A. Model Selection

Firstly , We selected 16 different CNN models commonly used in different tasks such as object detection ,semantic segmentation , image classification as backbone.These models vary from different block architecture , MAC count and parameter size.

B. Evaluation Metrics

To analyze the performance of each model, we defined a set of key metrics

- **Multiply-Accumulate Operations(MACs)**: Measures computational complexity .
- **Parameter Count**: In a given layer is the count of learnable elements for a filter.
- **Accuracy**: Measure of how well a neural network can predict outcomes.
- **Latency**: The time it takes for a model to make a prediction.
- **Power Consumption**: Energy used per inference.

C. Experimental Setup

The experiments were conducted using the following hardware and software configurations:

Hardware Platforms:

- **RTX 2060**
- **RTX 3070**

Software Tools:

- **NVIDIA Management Library (NVML)**: To track power consumption on NVIDIA GPUs.
- **PyTorch Framework**: Used for training and inference of models.

Dataset: The CIFAR-10 dataset consisting of 60000 32x32 colour images in 10 classes, with 6000 images per class, was chosen as the benchmark dataset. The training-to-testing data split was set at 80%-20%.

C. Power Consumption and Performance Measurement

To provide consistency in measurements , a standardized protocol was followed:

1) Each model was run for 30 times to inference on all hardware platforms.
2) Power consumption was measured continuosly over multiple inference runs to ensure stable operation.
3) Latency and accuracy metrics were recorded in parallel with power consumption measurements.
4) MAC count and parameter size is measured beginning of the inference.
5) Average power consumption , accuracy and latency per inference was calculated for analysis

D. Data Analysis and Comparison

The collected data was analyzed using the following methods:

- Relationships between power consumption and MAC count , parameter count , latency and accuracy were visualized..
- Differences between models for RTX 2060 and RTX 3070 were analyzed.

E. Findings and Insights

Finally , the most suitable model for each hardware platform was recommended. These recommendations aim to balance power consumption latency , and accuracy based on the specific requirements of applications, such as autonomous vehicles.

IV. Results

In this section, we present the results of the experiments conducted on two different hardware platforms, the NVIDIA RTX 2060 and NVIDIA RTX 3070. We evaluated the performance of 16 different Convolutional Neural Network (CNN) models, with a focus on power consumption, latency, accuracy, MAC count, and parameter size.

A. MAC Count vs. Power Consumption

A clear relationship was observed between the MAC count and power consumption. Models with more MAC operations consumed more power on both GPUs, but the RTX 3070 showed a higher increase in power consumption due to its larger computational resources

Model	MACs (G)	AVG GPU Power (W)
ConvNext	2.80	110.13
DenseNet	0.36	75.07
EfficientNetB0	0.03	31.83
EfficientNetB1	0.052	35.32
EfficientNetB2	0.059	34.29
EfficientNetB3	0.08	38.14
EfficientNetB4	0.13	50.06
EfficientNetB5	0.20	77.14
EfficientNetB6	0.29	85.55
EfficientNetB7	0.44	102.97
MobileNetV2	0.027	39.16
MobileNetV3	0.022	30.01
ResNet18	0.14	39.90
ShuffleNet	0.013	21.62
SqueezeNet1_0	0.05	31.92
SqueezeNet1_1	0.02	37.84

TABLE I
MACs vs Average GPU Consumption on RTX 2060.

B. Power Consumption vs. Parameter Count

The relationship between power consumption and parameter count was evaluated for both the RTX 2060 and RTX 3070. For both GPUs, models with a higher parameter count generally exhibited higher power consumption.

Model	MACs (G)	AVG GPU Power (W)
ConvNext	2.808	196.50
DenseNet	0.360	76.23
EfficientNetB0	0.035	34.84
EfficientNetB1	0.05	39.22
EfficientNetB2	0.059	40.29
EfficientNetB3	0.086	41.64
EfficientNetB4	0.133	48.81
EfficientNetB5	0.207	58.35
EfficientNetB6	0.294	84.21
EfficientNetB7	0.449	103.55
MobileNetV2	0.027	47.83
MobileNetV3	0.022	27.53
ResNet18	0.149	43.09
ShuffleNet	0.013	25.98
SqueezeNet1_0	0.053	33.65
SqueezeNet1_1	0.021	43.21

TABLE II
MACs vs Average GPU Consumption on RTX 3070.

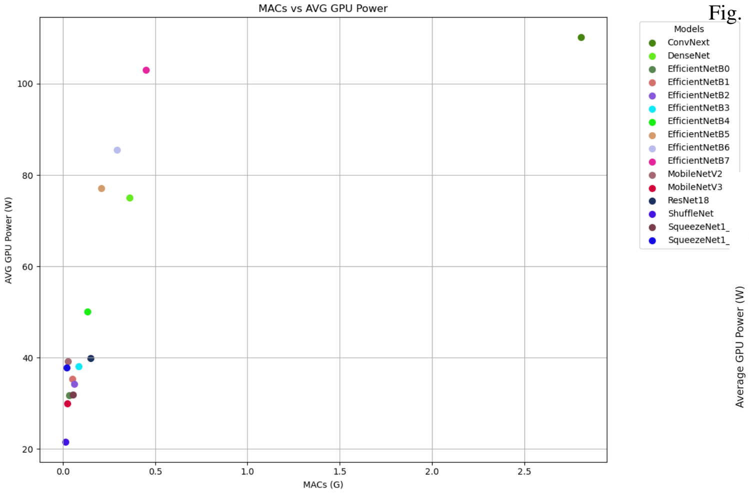

Fig. 1. MAC vs GPU power consumption for each model on RTX 2060.

Model	Params (M)	AVG GPU Power (W)
ConvNext	197.73	110.13
DenseNet	20.01	75.07
EfficientNetB0	5.28	31.83
EfficientNetB1	7.79	35.32
EfficientNetB2	9.10	34.29
EfficientNetB3	12.23	38.14
EfficientNetB4	19.34	50.06
EfficientNetB5	30.38	77.14
EfficientNetB6	43.04	85.55
EfficientNetB7	66.34	102.97
MobileNetV2	3.50	39.16
MobileNetV3	5.48	30.01
ResNet18	11.68	39.90
ShuffleNet	2.27	21.62
SqueezeNet1_0	1.24	31.92
SqueezeNet1_1	1.23	37.84

TABLE III
Parameter Count vs Average GPU Consumption on RTX 2060.

C. *Inference Time vs. Accuracy*

Contrary to initial expectations, the analysis revealed no clear correlation between latency and accuracy for the evalu-

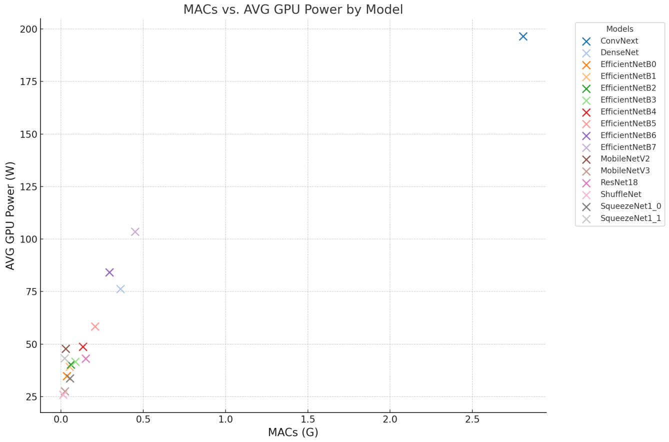

Fig. 2. MAC vs GPU power consumption for each model on RTX 3070.

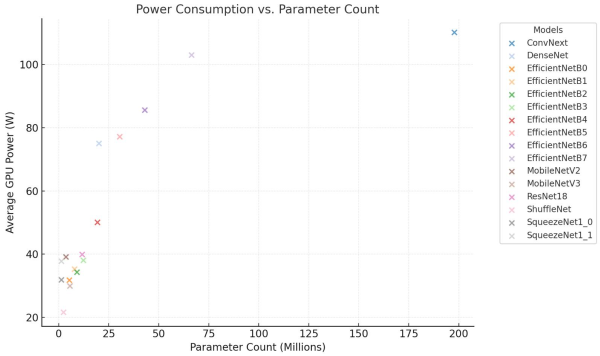

Fig. 3. Power Consumption vs Parameter Count on RTX 2060

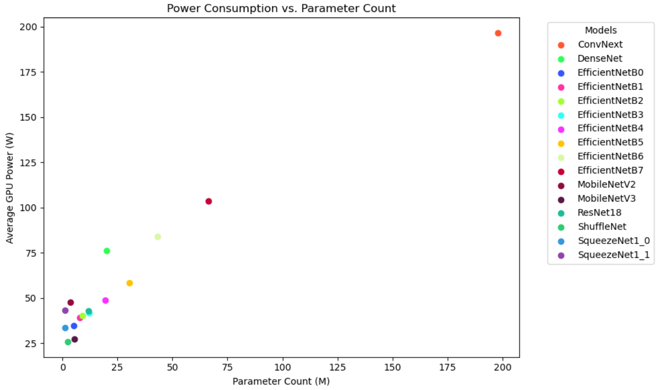

Fig. 4. Power Consumption vs Parameter Count RTX3070

Model	**Params (M)**	**AVG GPU Power (W)**
ConvNext	197.73	196.50
DenseNet	20.01	76.23
EfficientNetB0	5.28	34.84
EfficientNetB1	7.79	39.22
EfficientNetB2	9.10	40.29
EfficientNetB3	12.23	41.64
EfficientNetB4	19.34	48.81
EfficientNetB5	30.38	58.35
EfficientNetB6	43.04	84.21
EfficientNetB7	66.34	103.55
MobileNetV2	3.50	47.83
MobileNetV3	5.48	27.53
ResNet18	11.68	43.09
ShuffleNet	2.27	25.98
SqueezeNet1_0	1.24	33.65
SqueezeNet1_1	1.23	43.21

TABLE IV
PARAMETER COUNT VS AVERAGE GPU CONSUMPTION ON RTX 3070.

ated CNN models

Model	Inference Time (ms)	Accuracy (%)
ConvNext	18744.07	85.95
DenseNet	19917.87	86.35
EfficientNetB0	9116.52	89.57
EfficientNetB1	10965.62	89.83
EfficientNetB2	11324.40	90.12
EfficientNetB3	12568.85	89.71
EfficientNetB4	14313.74	89.97
EfficientNetB5	16751.70	88.89
EfficientNetB6	18523.79	89.48
EfficientNetB7	23370.25	89.31
MobileNetV2	7408.29	83.15
MobileNetV3	8036.33	86.14
ResNet18	5597.38	83.95
ShuffleNet	8740.48	84.85
SqueezeNet1_0	4884.71	75.06
SqueezeNet1_1	5158.07	78.23

TABLE V
INFERENCE TIME VS ACCURACY ON RTX 2060

Model	Inference Time (ms)	Accuracy (%)
ConvNext	10069.86	85.94
DenseNet	20661.98	86.35
EfficientNetB0	9062.46	89.56
EfficientNetB1	11214.32	89.81
EfficientNetB2	11304.82	90.12
EfficientNetB3	13184.08	89.71
EfficientNetB4	14501.89	89.98
EfficientNetB5	17404.02	88.87
EfficientNetB6	19062.21	89.48
EfficientNetB7	21917.11	89.33
MobileNetV2	6422.39	83.15
MobileNetV3	7703.23	86.17
ResNet18	4378.31	83.95
ShuffleNet	7670.21	84.85
SqueezeNet1_0	4099.06	75.05
SqueezeNet1_1	4158.53	78.24

TABLE VI
INFERENCE TIME VS ACCURACY ON RTX 3070

D. MAC Count vs. Inference Time

The number of Multiply-Accumulate operations (MACs) is directly proportional to inference time except ConvNext.

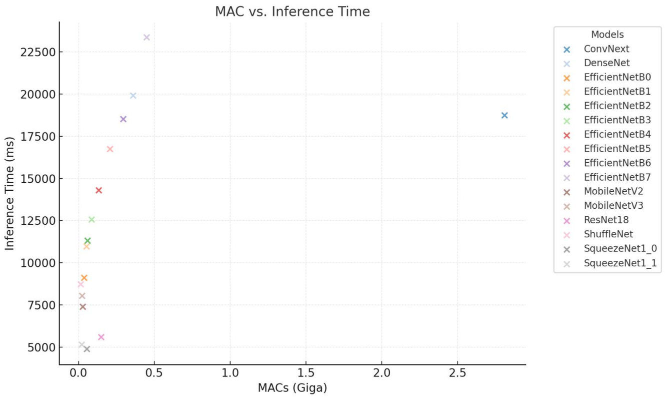

Fig. 5. MACs vs Inference Time on RTX2060

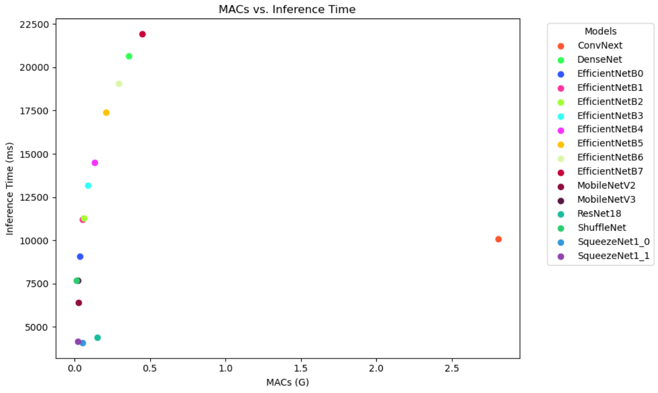

Fig. 6. MAC vs Inference Time on RTX3070

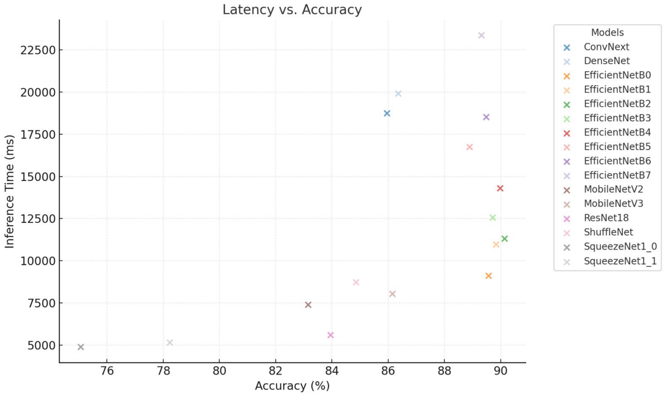

Fig. 7. Inference Time vs Accuracy on RTX2060

Model	MACs (G)	Inference Time (ms)
ConvNext	2.808	18744.07
DenseNet	0.360	19917.87
EfficientNetB0	0.0356	9116.52
EfficientNetB1	0.052	10965.62
EfficientNetB2	0.059	11324.40
EfficientNetB3	0.086	12568.85
EfficientNetB4	0.133	14313.74
EfficientNetB5	0.207	16751.70
EfficientNetB6	0.294	18523.79
EfficientNetB7	0.449	23370.25
MobileNetV2	0.027	7408.29
MobileNetV3	0.022	8036.33
ResNet18	0.149	5597.38
ShuffleNet	0.013	8740.48
SqueezeNet1_0	0.053	4884.71
SqueezeNet1_1	0.021	5158.07

TABLE VII
INFERENCE TIME VS MACS ON RTX 2060

Model	MACs (G)	Inference Time (ms)
ConvNext	2.808	10069.86
DenseNet	0.360	20661.98
EfficientNetB0	0.035	9062.46
EfficientNetB1	0.052	11214.32
EfficientNetB2	0.059	11304.82
EfficientNetB3	0.086	13184.08
EfficientNetB4	0.133	14501.89
EfficientNetB5	0.207	17404.02
EfficientNetB6	0.294	19062.21
EfficientNetB7	0.449	21917.11
MobileNetV2	0.027	6422.39
MobileNetV3	0.022	7703.23
ResNet18	0.149	4378.31
ShuffleNet	0.013	7670.21
SqueezeNet1_0	0.053	4099.06
SqueezeNet1_1	0.021	4158.53

TABLE VIII
INFERENCE TIME VS MACS ON RTX 3070

E. *RTX 2060 vs RTX 3070*

Despite differences in absolute performance metrics, the general behavior of the evaluated convolutional neural network (CNN) models on RTX 2060 and RTX 3070 was found to be consistent.

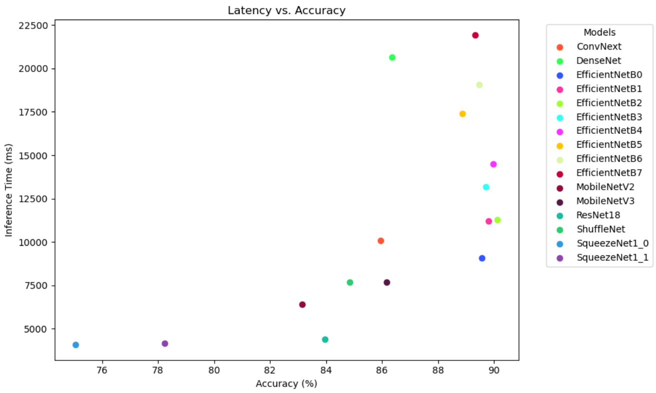

Fig. 8. Inference Time vs accuracy on RTX3070

V. DISCUSSION

The findings from our experiments provide significant insights into the relationship between power consumption, latency, accuracy, MAC count, and parameter size across two different hardware platforms: RTX 2060 and RTX 3070.

A. *MAC Count vs. Power Consumption*

Our results clearly show a direct correlation between MAC count and power consumption. Models with higher MAC counts, such as ConvNext, consume significantly more power compared to lightweight models like MobileNetV2 and ShuffleNet. This is because higher MAC operations require more computational resources, leading to increased power usage. For the RTX 3070, the power consumption rises more steeply with increasing MAC count due to its greater computational capacity. In contrast, the RTX 2060, while consuming less power overall, still shows an increase in power consumption with higher MAC counts but at a slower rate. This suggests that the RTX 2060 is more energy efficient for models with higher MAC numbers, which could be useful in power-constrained environments where inference time is not a concern (see table IX).

B. *Power Consumption vs Parameter Count*

There is a clear correlation between parameter count and power consumption, as shown in figure and 4. As the number of parameters in a model increases, power consumption tends to rise as well. For example, EfficientNetB7 and ConvNext, both of which have a high parameter count, consume significantly more power compared to models like MobileNetV3 and ShuffleNet, which have fewer parameters.

C. *MAC Count vs. Inference Time*

The relationship between MAC count and inference time is explored in Figure 5 and Figure 6.In general, as the number of macs increases, the inference time increases, but ConvNext made an exception here, the main reason for this may be that it has fewer activations and the ConvNext architecture uses cuda cores more efficiently.While this behavior is more pronounced for the RTX 3070, the behavior of other models is parallel to the RTX 2060.

D. *RTX 2060 vs RTX 3070*

A comparison between the RTX 2060 and RTX 3070 reveals noticeable differences in inference time and average gpu power , especially in higher MAC counts , as shown in Table IX

VI. CONCLUSION

In this study, we explored the relationship between power consumption, MAC count, and parameter size for various Convolutional Neural Networks (CNNs) using RTX 2060 and RTX 3070 GPUs. By evaluating 16 different CNN models on the CIFAR-10 dataset, we examined how architectural design, hardware, and model complexity influence key performance indicators such as accuracy, power consumption, inference time, and MAC operations.

Our findings indicate that there is a direct correlation between the MAC count and power consumption, highlighting that more computationally demanding models tend to consume more power. Interestingly, we observed that the GPU used for training (RTX 2060 vs. RTX 3070) did not significantly impact the accuracy of the models, suggesting that accuracy may be less dependent on hardware and more dependent on the model architecture itself. However, the model architecture was found to have a significant effect on inference time, with more complex architectures leading to longer processing times despite having similar accuracy levels.

Based on these insights, we conclude that optimizing for power efficiency in embedded systems, such as autonomous vehicles and robots, involves balancing the MAC count, number of parameters, and inference time. For tasks like object detection or image classification in resource-constrained environments, lightweight CNN models with a favorable balance of low power consumption and high accuracy should be prioritized. The research also underscores the importance of model selection for specific hardware, with the RTX 2060 and RTX 3070 exhibiting differences in power consumption but similar accuracy levels across the models tested.

In future work, further optimization can be achieved by exploring advanced CNN architectures or alternative techniques such as pruning, quantization, or hardware-specific optimizations. The goal is to design energy-efficient models suitable for real-time, low-power applications without compromising performance.

Acknowledgment

The author would like to thank Tolga Kurtuluş Çapın

References

[1] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[2] A. Howard, M. Sandler, G. Chu, L. C. Chen, B. Chen, M. Tan, *et al.*, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 1314–1324, doi: 10.1109/ICCV.2019.00140.

[3] S. Mittal, "A survey on optimized implementation of deep learning models on the NVIDIA Jetson platform," *J. Syst. Archit.*, vol. 97, pp. 428–442, 2019, doi: 10.1016/j.sysarc.2019.10.006.

[4] C. Zhang, P. Li, X. Qin, and Y. Liu, "Efficient CNN inference on edge devices with low memory requirements," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5315–5329, 2020, doi: 10.1109/TNNLS.2020.2986179.

[5] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 28, pp. 1135–1143, 2015.

[6] A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," *arXiv preprint*, arXiv:1605.07678, 2016, doi: 10.48550/arXiv.1605.07678.

[7] T. J. Yang, A. Howard, B. Chen, X. Zhang, A. Go, V. Sze, and H. Adam, "NetAdapt: Platform-aware neural network adaptation for mobile applications," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2017, pp. 285–300, doi: 10.1007/978-3-030-01216-8_18.

[8] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html.

[9] Z. Chen, L. Fang, J. Xu, and Z. Liu, "Low-power CNNs for efficient object detection in autonomous systems," *IEEE Access*, vol. 8, pp. 130497–130510, 2020, doi: 10.1109/ACCESS.2020.3008866.

...

Model	Inference Time Difference (ms)	Accuracy Difference (%)	AVG GPU Power Difference (W)
ConvNext	-8674.20	-0.01	86.36
DenseNet	744.11	0.0	1.16
EfficientNetB0	-54.06	-0.009	3.01
EfficientNetB1	248.70	-0.019	3.89
EfficientNetB2	-19.58	0.0	6.00
EfficientNetB3	615.23	0.0	3.49
EfficientNetB4	188.15	0.01	-1.25
EfficientNetB5	652.32	-0.01	-18.78
EfficientNetB6	538.41	0.0	-1.34
EfficientNetB7	-1453.13	0.019	0.57
MobileNetV2	-985.89	0.0	8.67
MobileNetV3	-333.09	0.03	-2.48
ResNet18	-1219.06	0.0	3.19
ShuffleNet	-1070.27	0.0	4.36
SqueezeNet1_0	-785.64	-0.01	1.73
SqueezeNet1_1	-999.53	0.009	5.36

TABLE IX

RTX 3070 AND RTX 2060 DIFFERENCE