# Energy Consumption Analysis of Convolutional Neural Networks

**Xiaokun Qi, Tian He***

*College of Mechanical and Electrical Engineering, Qingdao University, Qingdao, 266071, China*
*qixiaokun1210@foxmail.com*
**Corresponding author: he_t@x263.net*

***Abstract:*** *The rapid development and widespread application of convolutional neural networks have played a positive role in promoting social progress and economic development. An increasing number of neural networks are being deployed on mobile devices to meet the needs of visual perception tasks. However, the operation of complex neural networks requires a large amount of computational cost, which affects the working duration of mobile devices. Therefore, it becomes crucial to analyze the energy consumption of neural networks. Through energy consumption analysis, the main energy consumption points can be identified, providing ideas for the subsequent low-energy consumption of neural networks. This paper selects six classic convolutional neural networks, AlexNet, VGG, GoogLeNet, ResNet, MobileNet, and ShuffleNet, to study the impact of different network structures on energy consumption, and compares their running time, power, and energy consumption. On this basis, the hotspot layers (convolutional layer, pooling layer, fully connected layer) are analyzed, and it is found that the energy consumption of the convolutional layer accounts for the largest proportion. It is inferred from this that the internal structure of the convolutional layer is a key factor affecting the energy consumption of neural networks. Based on this, improvements to the internal structure of the convolutional layer can reduce algorithm energy consumption.*

***Keywords:*** *Convolutional Neural Networks, Hotspot Layer, Energy Consumption Analysis*

## 1. Introduction

Due to their powerful feature representation and generalization capabilities, an increasing number of neural networks are being deployed on mobile devices such as smartphones, cameras, and wearable devices. The deployment of neural network models on mobile devices provides users with more intelligent functions and experiences, such as facial recognition, voice assistants, and image filters. However, these mobile devices are usually small in size and rely on battery power. Limited by internal space and battery capacity, the available energy is limited. When neural networks perform tasks on mobile devices, their design involves a large amount of computation, which requires high energy consumption. For example, ResNet-50, which has 50 convolutional layers, needs to occupy more than 95MB of memory and perform more than 3.8 billion floating-point multiplications during the inference phase of image processing[1]; AlexNet[2], the basic network for image classification, runs out of all phone power in less than an hour[3]. The problem of energy consumption limitation of mobile neural networks is becoming increasingly prominent.

The problem of energy consumption limitation greatly affects the running time of the device, shortens the service cycle, reduces the efficiency of use, and hinders the application of mobile devices in daily life. In order to overcome these difficulties, the energy consumption performance of neural networks is evaluated at the algorithm level, and the energy consumption problem of neural network models during operation is explored. In this work, we first compare the time, power, and energy consumption of classic convolutional neural networks such as AlexNet[2], VGG[4], GoogLeNet[5], ResNet[6], MobileNet[7], ShuffleNet[8] when performing the same tasks, then analyze the impact of different hotspot layers (convolutional layer, pooling layer, fully connected layer) on runtime energy consumption, and finally determine the key factors affecting energy consumption.

The rest of this paper is arranged as follows. Section 2 reviews related research, Section 3 analyzes the energy consumption of six classic neural networks, and Section 4 concludes.

## 2. Related Research

Energy issues have always been a long-term concern for countries around the world, and it is extremely important to do a good job in energy work throughout the year. In earlier research, Han[9] analyzed convolutional neural network models from a holistic perspective. They extracted the number of Multiply-Accumulate (MAC) operations and the number of weights from pre-trained deep learning models, simulated the number of floating-point operations of the processor by calculating the number of MAC operations, and used the number of weights to simulate the number of main memory accesses. They associated known costs with each type of operation to determine the energy consumption of deep learning models. Rouhani[10] analyzed the energy consumption during the forward propagation phase of neural networks in the training stage. The authors modeled energy use based on basic arithmetic operations and shared weight communication in distributed training settings. At the same time, the authors found through experiments that in some cases, appropriately increasing the size of the convolution kernel in the initial stage can reduce inference time and achieve the purpose of reducing energy consumption. In subsequent related research, Chen[11] argued that using the number of weights for energy consumption analysis was overly simplistic and could not effectively reflect the overall energy consumption of the network. Therefore, they attempted to incorporate the energy costs of different types of data at different memory levels. By using the number of times each data value is reused in the memory hierarchy during the computation process, a comprehensive energy consumption evaluation of the model is conducted. The introduced four-level memory structure includes DRAM, global buffer, array, and register file, each with corresponding energy consumption. Other scholars have considered the energy consumption of computational operations and communication. Qi.[12] focused on analyzing the running time of neural networks, breaking down the running time of neural networks to calculate each layer separately, and estimating the overall energy consumption in combination with the rated power of the hardware. This method estimates the execution time of deep learning systems by mapping the computational requirements of the neural network architecture to the design space of software, hardware, and communication strategies. Energy consumption estimation can then be associated with the execution time and the power parameters of the hardware, thereby deriving the energy consumption estimate of the deep learning system. In terms of computation time, factors such as the computational input size of the network architecture, the algorithms and operations involved in the network layer, and the performance of the hardware are considered. By considering these factors comprehensively, the consumption of computation time can be estimated. In terms of communication time, factors such as the computational dependencies in the network, the communication bandwidth of the hardware, and parallelization schemes are considered. Through the analysis of these factors, the consumption of communication time can be estimated.

Since data movement occupies a significant part of energy consumption, it is difficult to directly extract energy consumption from the neural network model. Yang [13] conducted an energy consumption analysis based on the architecture, sparsity, and bit width of the neural network. The authors believe that data movement accounts for most of the energy consumption, while the energy consumption of the neural network during computational operations is relatively less. The energy consumption of computation can be approximated by estimating the number of multiply-add operations, while the energy consumption of data movement needs to consider the energy consumption of memory access. When analyzing energy consumption, it is necessary to consider the access times and energy consumption of different levels of memory to estimate the energy consumption of data movement. The authors also introduced a normalized energy cost, comparing the energy consumption of memory access with the energy consumption of multiply-add operations. The authors also provide an online energy estimation tool, aiming to bridge the gap between algorithm and hardware design, and provide useful insights for the development of energy efficient DNNs. In order to study the energy consumption of the model more specifically, Lu[14] focused on analyzing the execution time and resource usage of only the convolutional layer in the convolutional network model. They regard matrix multiplication as the main component of the convolutional layer and perform different sizes of matrices separately to simulate their performance and resource usage. However, this method has a clear disadvantage, approximating performance as isolated matrix multiplication operations cannot capture dependencies, including memory reuse that occurs between layers during actual inference runtime. Therefore, compared with the actual measured execution time, this method often overestimates the execution time of each layer. Cai[15] proposed to use statistical regression techniques to analyze the power consumption and running time of GPUs at the algorithm level. Unlike the above methods, this method takes the convolutional layer, pooling layer, and fully connected layer as the research objects, and uses the actual power and timing values to indirectly reflect the average power and running time. The authors exclude the influence of voltage and frequency scaling by keeping the GPU in a fixed state, take the convolution kernel size, network layers, etc., as the input of the model, combine the output of the power prediction model and the runtime model, and give an estimate of the energy consumption of each layer. Then add the estimates of each layer to get the energy consumption estimate of the entire convolutional neural network.

For more accurate energy consumption analysis, Faviola[16] expanded on the basis of their predecessors. In order to conduct a more comprehensive energy consumption analysis, the authors used two schemes: analysis based on a single layer and analysis based on layer type. The analysis based on a single layer uses complex features, such as extracting input dimensions, convolution kernel size, stride, padding, etc., from different structural layers for energy consumption analysis. The analysis based on layer type uses relatively simple features, such as the sum of multiply-accumulate (MAC) operations, for energy consumption analysis. By comparing the results of these two types, the authors found that the analysis using layer type features is comparable in accuracy to using more complex features, but the complexity is reduced by 4 to 32 times. This means that using simple layer type features can reduce the complexity of the model while maintaining accuracy, thereby having higher efficiency and interpretability.

## 3. Energy Consumption Analysis

### *3.1 Energy Consumption Comparison*

During the development process, the algorithm model will directly affect the system’s energy consumption. By conducting an energy consumption analysis of the algorithm, we can better understand the differences in energy consumption among different algorithms. This helps us select and design more energy-efficient algorithms to reduce energy consumption during computation. The internal structure of different algorithms will lead to different patterns and efficiencies of energy consumption. To explore their internal impact, this section selects six classic convolutional neural networks for energy consumption analysis, trying to seek the impact of different module layouts on overall energy consumption. The selected classic convolutional neural networks are AlexNet, VGG, GoogLeNet, ResNet, MobileNet, and ShuffleNet.

Firstly, we respectively conduct training time statistics for the above six classic convolutional neural networks on the GPU, with each neural network training for 50 epochs. Of course, during training, in order to maintain the same experimental conditions, we will limit the size of the input image and use the same size batch, only maintaining the difference of the neural network model itself. In order to better reflect the actual running situation, the total running time we test here includes network initialization and data preparation time. However, the analysis of running time alone is far from enough to fully evaluate the energy consumption problem. In order to more comprehensively consider the factors of energy consumption, in the following experiments, we also obtained the average power data of each model during operation. This can better reflect the energy consumption characteristics of different models in actual operation. As we all know, it is meaningless to discuss the running time or running power of a model alone. In the field of computer science and engineering, we need to conduct a comprehensive energy consumption analysis and consider multiple factors. Just focusing on running time may ignore other key factors that affect energy consumption. For example, a model may run in a short time, but the power it requires may be very high, leading to a large overall energy consumption. On the contrary, another model may require more running time, but the power consumption is low, so it is more efficient in terms of energy consumption. Similarly, just focusing on running power cannot fully evaluate the energy consumption problem. A model may have a lower average power, but its running time is longer, resulting in a higher total energy consumption. Another model may have a higher power, but the running time is short, so it is more energy-saving in terms of total energy consumption. Therefore, we need to consider both running time and running power to get a more accurate energy consumption evaluation. Only after a comprehensive analysis of these factors can we draw more practical conclusions and provide targeted solutions for energy consumption optimization. Therefore, after obtaining the running time and average power of a single model, we can calculate its total energy consumption through the formula, as shown in equation (1):

$$E = \bar{P}\,T \tag{1}$$

This formula combines the two factors of running time and average power to give the total energy consumed by the model during operation. By calculating the total energy consumption, we can more comprehensively evaluate the energy consumption of different models. This helps us compare the energy consumption differences between different models and provide references for energy consumption optimization. We can choose those models that have a lower average power but a moderate running time, thereby reducing the total energy consumption while meeting the task requirements. This method of considering both running time and average power can help us more accurately evaluate and compare the energy efficiency of models. By optimizing the total energy consumption of the model, we can achieve more efficient computing and more sustainable development.

We compared the total running time, power, and total energy consumption of six models, and the results are shown in Figure (1).

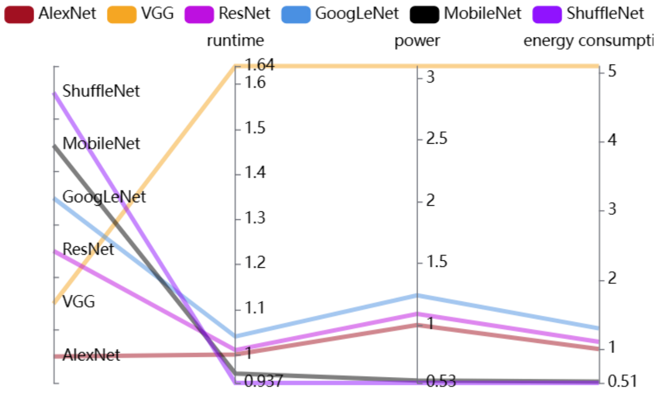

*Figure 1: Comparison of Running Time, Power, and Energy Consumption during Training of Classic Neural Network Models*

Similarly, we also analyze the running time, average power, and total energy consumption during inference. The results are shown in Figure (2).

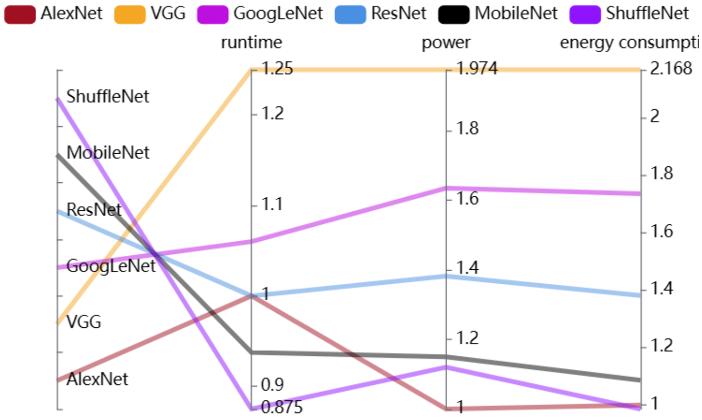

*Figure 2: Comparison of Running Time, Power, and Energy Consumption during Inference for Classic Neural Network Models*

From the data in Figures (1) and (2), we can observe that apart from the VGG model, the running time differences of other models are not particularly significant. However, there are noticeable differences in their running power, which directly affects their total energy consumption, further illustrating the necessity of multi-angle energy consumption analysis. In addition, we can also observe that with the development of model lightweighting, there will be a reduction in the total running time of the model, and its running power is also decreasing. This implies that the lightweighting of models has promoted the energy optimization of neural network models. From this, we can infer that there may exist some kind of lightweighting method or some convolution operation that can greatly reduce the energy consumption of the model. Based on these observations, we can infer that there may exist some kind of lightweighting method or some convolution operation that can significantly reduce the energy consumption of the model. This discovery provides insights for our further research and development of energy consumption issues.

By comprehensively analyzing running time, running power, and total energy consumption, we can evaluate the energy efficiency of different models. This comprehensive energy consumption analysis method helps us better understand the energy consumption characteristics of models, providing a scientific basis for energy conservation, emission reduction, and sustainable development.

### *3.2 Hotspot Layer Analysis*

By conducting a hotspot layer analysis on CNN (Convolutional Neural Networks), we can better understand and interpret the key layers within the network. This analysis can help us comprehend the model's representational power and feature extraction capabilities at different levels of abstraction. Specifically, by analyzing the hotspot layers at different levels, we can identify which layers are more important for specific tasks. This can assist us in focusing on and enhancing the performance of these key layers when designing and optimizing the model. Additionally, hotspot layer analysis can also help us examine the runtime and energy consumption levels of different layers. By analyzing the impact of the internal structure of different layers on energy consumption, we can identify layers that may lead to higher energy consumption and optimize accordingly. By considering the runtime or energy consumption levels of layers, we can conduct a comprehensive energy analysis of the CNN model. This helps us understand the model's characteristics in terms of energy consumption and provides guidance for energy optimization.

We have decomposed the six classic convolutional neural network models mentioned above, namely AlexNet, VGG, GoogLeNet, ResNet, MobileNet, and ShuffleNet, to collect energy consumption information for each layer. These models are mainly divided into three levels: convolutional layers, pooling layers, and fully connected layers. We know that in addition to these layer structures, there are other structures in the neural network model, such as Batch Normalization (BN) layers, activation function layers, and loss function layers, but we choose to discard these layers. However, when I say 'discard', it does not mean they do not participate in the calculation. Since they usually follow the convolutional layers, for the sake of convenient calculation, we include them in the statistics of the convolutional layers. If we maliciously decompose the continuous structure, it will affect the integrity of the algorithm structure. The experimental results are shown in Figure 3.

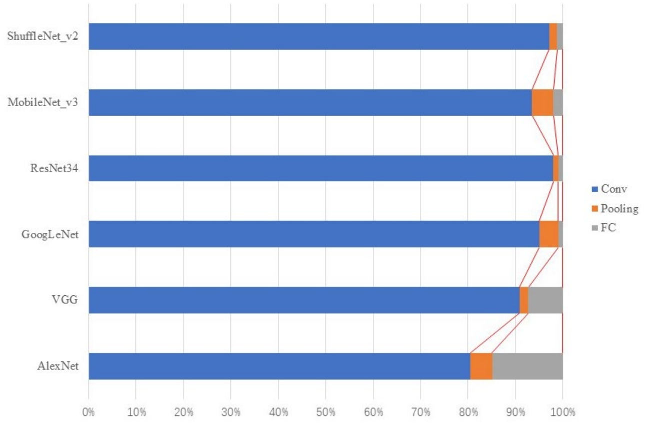

*Figure 3: Proportional Energy Consumption of Hotspot Layers in Classic Convolutional Neural Networks*

From the information shown in the figure above, it is clear that the convolutional layer consumes the most energy, while the pooling and fully connected layers vary depending on the model. This is because the convolutional layer is often one of the parts with the largest computational load in neural network learning models. Convolution operations involve a large number of multiplication and addition operations, so they consume more energy. This is why the energy consumption of the convolutional layer is usually higher than other layers. The energy consumption of the pooling and fully connected layers will vary depending on the specific model architecture and parameter settings. The pooling layer mainly performs downsampling on the feature map, reducing the number of parameters and computations, so its energy consumption is relatively low. The fully connected layer needs to compute all input nodes, so its energy consumption will increase with the number of input nodes. By observing and analyzing the energy consumption differences of different layers, we can better understand the contribution of different layers to energy consumption in deep learning models. This has important guiding significance for energy optimization and model design. We can explore some optimization methods for the high energy consumption of the convolutional layer, such as network pruning and quantization, to reduce the amount of computation. At the same time, for the pooling layer and fully connected layer, we can make appropriate adjustments and designs according to specific needs and performance requirements.

## 4. Conclusion

From this experiment, it is clear that although many different convolutional neural network models have different internal structures, without exception, the convolutional layer is the part of the neural network that consumes the most energy during operation. In comparison, the energy consumption of the pooling layer and the fully connected layer is not as significant. The reason why the convolutional layer has such high energy consumption is because it contains a large number of learning parameters, involves a large number of addition and multiplication operations, and occupies a large amount of computing resources. Therefore, if researchers want to involve low-energy neural networks, they can start with the configuration of the convolutional layer to reduce the overall energy consumption.

## References

*[1] CHENG Y, WANG D, ZHOU P, et al. A Survey of Model Compression and Acceleration for Deep Neural Networks [J]. ArXiv, 2017, abs/1710.09282.*

*[2] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks [J]. Communications of the ACM, 2012, 60: 84 - 90.*

*[3] YANG T-J, CHEN Y-H, SZE V. Designing Energy-Efficient Convolutional Neural Networks Using Energy-Aware Pruning [J]. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: 6071-9.*

*[4] SIMONYAN K, ZISSERMAN A. Very Deep Convolutional Networks for Large-Scale Image Recognition [J]. CoRR, 2014, abs/1409.1556.*

*[5] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions [J]. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014: 1-9.*

*[6] HE K, ZHANG X, REN S, et al. Deep Residual Learning for Image Recognition [J]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015: 770-8.*

*[7] HOWARD A G, SANDLER M, CHU G, et al. Searching for MobileNetV3 [J]. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019: 1314-24.*

*[8] MA N, ZHANG X, ZHENG H, et al. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design [J]. ArXiv, 2018, abs/1807.11164.*

*[9] HAN S, LIU X, MAO H, et al. EIE: Efficient Inference Engine on Compressed Deep Neural Network [J]. 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), 2016: 243-54.*

*[10] ROUHANI B D, MIRHOSEINI A, KOUSHANFAR F. DeLight: Adding Energy Dimension To Deep Neural Networks [Z]. Proceedings of the 2016 International Symposium on Low Power Electronics and Design. San Francisco Airport, CA, USA; Association for Computing Machinery. 2016: 112–7.10.1145/2934583.2934599*

*[11] CHEN Y-H, EMER J, SZE V. Eyeriss: a spatial architecture for energy-efficient dataflow for convolutional neural networks [J]. SIGARCH Comput Archit News, 2016, 44(3): 367–79.*

*[12] QI, SPARKS E R, TALWALKAR A. Paleo: A Performance Model for Deep Neural Networks; proceedings of the International Conference on Learning Representations, F, 2016 [C].*

*[13] YANG T-J, CHEN Y-H, EMER J, et al. A method to estimate the energy consumption of deep neural networks; proceedings of the 2017 51st asilomar conference on signals, systems, and computers, F, 2017 [C]. IEEE.*

*[14] LU Z, RALLAPALLI S, CHAN K S, et al. Modeling the Resource Requirements of Convolutional Neural Networks on Mobile Devices [J]. Proceedings of the 25th ACM international conference on Multimedia, 2017.*

*[15] CAI E, JUAN D-C, STAMOULIS D, et al. NeuralPower: Predict and Deploy Energy-Efficient Convolutional Neural Networks [J]. ArXiv, 2017, abs/1710.05420.*

*[16] FAVIOLA RODRIGUES C, RILEY G, LUJAN M. Energy Predictive Models for Convolutional Neural Networks on Mobile Platforms [J]. arXiv e-prints, 2020: arXiv: 2004.05137.*