# EPAM: A Predictive Energy Model for Mobile AI

Anik Mallik*, Haoxin Wang†, Jiang Xie*, Dawei Chen‡, and Kyungtae Han‡

*Department of Electrical and Computer Engineering, The University of North Carolina at Charlotte, NC, USA

†Department of Computer Science, Georgia State University, GA, USA

‡InfoTech Labs, Toyota Motor North America R&D, Mountain View, CA, USA

***Abstract*—Artificial intelligence (AI) has enabled a new paradigm of smart applications – changing our way of living entirely. Many of these AI-enabled applications have very stringent latency requirements, especially for applications on mobile devices (e.g., smartphones, wearable devices, and vehicles). Hence, smaller and quantized deep neural network (DNN) models are developed for mobile devices, which provide faster and more energy-efficient computation for mobile AI applications. However, how AI models consume energy in a mobile device is still unexplored. Predicting the energy consumption of these models, along with their different applications, such as vision and non-vision, requires a thorough investigation of their behavior using various processing sources. In this paper, we introduce a comprehensive study of mobile AI applications considering different DNN models and processing sources, focusing on computational resource utilization, delay, and energy consumption. We measure the latency, energy consumption, and memory usage of all the models using four processing sources through extensive experiments. We explain the challenges in such investigations and how we propose to overcome them. Our study highlights important insights, such as how mobile AI behaves in different applications (vision and non-vision) using CPU, GPU, and NNAPI. Finally, we propose a novel Gaussian process regression-based general predictive energy model based on DNN structures, computation resources, and processors, which can predict the energy for each complete application cycle irrespective of device configuration and application. This study provides crucial facts and an energy prediction mechanism to the AI research community to help bring energy efficiency to mobile AI applications.**

***Index Terms*—mobile AI, predictive energy model, energy improvement, latency reduction, DNN**

## I. Introduction

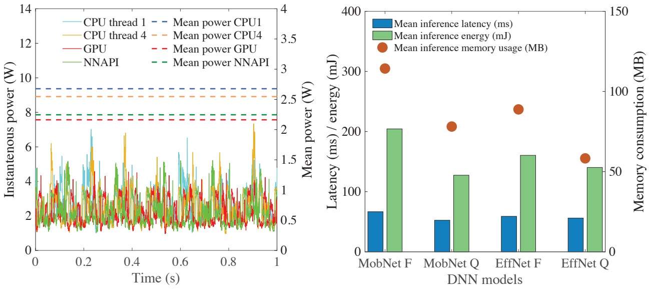
Artificial intelligence (AI) is shaping every aspect of human lives nowadays. Furthermore, mobile devices, i.e., smartphones, tablets, wearable devices, and autonomous and unmanned aerial vehicles, are heavily invested in AI applications, having cellular networks, edge, and cloud computing in the backbone. AI applications consume considerably high energy and memory of these devices. How AI uses these resources defines a device's potential to interact with wireless networks. Therefore, it is crucial to understand the characteristics of AI applications running on a mobile device, which pushes back to the question — how can we accurately predict the energy consumption of mobile AI irrespective of device configurations to ensure better service and user experience?

AI applications' energy consumption may depend on various properties of a system. First, the AI models that are crafted in specific ways to fit mobile devices due to the models' high computation and energy requirements, impact the applications' behaviors. Research works suggest accelerating the processing time of deep neural networks (DNNs) by quantizing [1], which is a compression technique run on DNN models that can reduce the model size by converting some tensor operations to integers from floating points or reducing the weights or parameters in a model, but at the cost of degraded accuracy. Quantized DNN (Q-DNN) models are generally investigated for vision-based applications, the most thriving areas of AI. Second, mobile AI is not limited to vision applications only. Modern-day mobile devices are rigged with non-vision applications as well, such as intelligent recommendations, natural language processing (NLP), smart reply, speech recognition, and speech-to-text conversion. While most of the research focuses on applications based on computer vision, acquiring a thorough knowledge of mobile AI is only possible by including non-vision applications. Third, the processing source used to run the AI models affects their performance. Besides central processing units (CPUs) with high processing speeds, some devices are now equipped with graphics processing units (GPUs), which enables DNN models to run faster than ever, especially for vision applications [2]. In addition, neural network application programming interfaces (NNAPI) are also developed to make the processing of DNN models faster using CPUs, GPUs, or neural processing units (NPUs) [3]. These state-of-the-art technologies are researched for mobile AI only to improve inference latency. Lastly, the hardware configuration of mobile devices is distinctive and contributes to energy consumption with a unique signature. The system-on-chip (SoC), CPU/GPU parameters, and memory dictate how an AI application runs on a specific device.

In this paper, we argue that a predictive energy model for a mobile AI application requires considering all of the parameters mentioned above. Without collecting accurate and precise latency, energy, and memory consumption data, it is not possible to design a predictive energy model which is applicable to all AI applications with different model sizes and device configurations. This paper presents the measurement data of AI applications collected through experiments and proposes a novel model of Energy Prediction for AI in Mobile devices (EPAM), which can provide a highly accurate prediction of the energy consumption of a mobile AI application irrespective of device configuration and AI models, and thus contribute to improving the overall performance.

**Motivations:** While mobile AI is often concluded as "no one-size-fits-all solution" [4], it is the responsibility of the research community to provide the developers with precise measurement data and a way to predict energy consumption. Our research shows that the power varies for the same device with the change in processing sources (Fig. 1(a)). The granularity of power consumption over a unit period of time needs to be measured to develop a predictive energy model, which is not provided by the current works. Battery profilers provided by third-party applications do not support precise energy data collection [5]. Hence, the use of an external power

This work was supported by funds from Toyota Motor North America and by the US National Science Foundation (NSF) under Grant No. 1910667, 1910891, and 2025284.



(a) (b)

Fig. 1. (a) Power consumption by different processors for the same time interval for MobileNet Float and (b) mean inference latency, energy, and memory usage for float and quantized DNN models on Huawei Mate40Pro.

measurement tool becomes necessary [6]. Moreover, DNN models with different sizes and layers do not have a similar impact on the latency, energy, and memory usage, which is presented in Fig. 1(b), where it is evident that the correlation among latency, energy, and memory is not linear at all. An interesting observation here is that the Quantized EfficientNet model causes high latency and energy despite using the lowest memory, due to its compatibility issues with NNAPI, which is described in detail in section V-A. This motivates us to collect data from a physical testbed to validate this correlation before proposing a predictive energy model.

**Challenges:** Designing a predictive energy consumption model for mobile AI is not straightforward. First, a general energy prediction model is challenging to develop *due to different categorical and numerical variables involved in the non-parametric behavior of the energy consumption of AI applications.* The regression model cannot be linear since all the parameters do not have the same weight in all applications and configurations. Second, measuring mobile AI parameters is challenging due to *complicated power terminal design in the latest mobile devices.* Synchronizing the timestamps of latency and energy data brings further difficulties as the retrieved log files have different formats. However, these parameters must be measured since they are required for training the regression model. Finally, the *experiments should be controllable and repeatable* for enthusiastic researchers. Therefore, the environment must be chosen wisely so that all the experiments can be carried out in a similar condition.

**Our contributions:** Our contributions in this paper are summarized as follows:

- **Experimental research and analysis of different mobile AI applications:** We set up an experimental testbed with four different smartphones (Table I) and use a vision application (image classification) and two non-vision applications (NLP and speech recognition) with seven different DNN models (Table II). The testbed is described in detail in Section IV. We investigate different mobile AI parameters through an extensive experimental study. The latency, power consumption, and memory usage of individual segments of the pipelines of three AI applications are measured for different applications using single- and multi-threads CPU, GPU, and NNAPI and for different DNN models. Our experiment shows that the total energy consumption of a mobile AI application is related to the device configuration, AI model, latency, and memory.
- **Predictive energy model for mobile AI:** We propose a novel Gaussian process regression-based general predictive energy model for mobile AI (EPAM) based on DNN structure, memory usage, and processing sources to predict the energy consumption of mobile AI applications irrespective of device configurations (Section III). EPAM requires offline training with past datasets. The trained model can be used to predict the overall energy consumption which reduces the necessity for further energy measurement and helps the developers design energy-efficient mobile AI applications. Finally, we evaluate the performance of our proposed predictive energy model EPAM with our experimental data (Section V-D). The evaluation shows that EPAM provides highly accurate energy prediction of vision and non-vision AI applications for different DNN models on unique mobile devices.

## II. Related Work

**Vision and non-vision mobile AI with float and quantized models:** Floating point and quantized models are investigated for vision applications, e.g., image classification, segmentation, super-resolution, and object detection, to create benchmarks using inference latency for mobile devices [7]. Quantized models are introduced in [8] to lower the energy consumption as well. In addition, non-vision AI applications are also researched to achieve high accuracy and low latency [9]. Nevertheless, a predictive energy model for mobile AI requires analysis of complete behaviors of vision and non-vision mobile AI applications using floating point and quantized models, which are not yet explored.

**Latency and energy in different processors:** Mobile AI applications behave differently in terms of latency and accuracy based on the processing sources [4], [7]. Research works are done on maximizing CPU threads [10] and hardware acceleration for DNN models. The use of GPU is also studied for improving the training and inference time for mobile AI [2]. NPU architectures are explored as well to expedite neural network operations [3], [11]. However, there is no fundamental framework to describe the impact of individual processing sources on energy consumption for different mobile AI applications with disparate DNN models.

**Energy modeling for mobile AI and prediction:** Energy measurement is necessary to describe mobile AI applications' detailed behaviors. Eprof [12] and E-Tester [13] are proposed to measure and test the battery drain of mobile devices, which use a finite state machine to measure the energy. However, these methods lack in providing granular and precise energy data since they only act on system call traces. Researchers have proposed different energy models for vision [14] and non-vision [15] applications. Furthermore, predictive energy models are developed for devices, and sensors [16]. Nonetheless, developing accurate predictive energy models general to all mobile AI applications requires knowledge of all the environmental parameters such as network and model size, memory usage, and the hardware accessed to run the AI application.

## III. EPAM: Overview of the Predictive Model

The energy prediction of mobile AI involves a high dimension of influencing variables, making it a non-parametric model. Let us assume that the set of input data points is $X^{1:D}$, where D is the total number of dimensions. If we consider this a noisy observation, then we find the posterior distribution as

$$P(E(X) \propto P(E(X)|\Lambda^{1:D})/P(\Lambda^{1:D}|E(X)), \tag{1}$$

where $E(X)$ is the observed energy at data points $X^{1:D}$ and $\Lambda^{1:D} = \{X^{1:D}, E\}$ is observation points. Using Gaussian process [17], $E(X)$ can be described as $E(X) \sim \mathcal{N}(\mu, K)$, where $\mu = [mean(X^1), \ldots, mean(X^D)]$ is the mean and $K_{ij} = k(x_i, x_j)$ is the covariance or Kernel function, where $x_i$ and $x_j$ are distinct data points.

As new data points $X_*$ are provided, the posterior distribution of predicted energy $E(X_*)$ can be modeled as

$$P(E(X_*)|\Lambda^{1:D}) \sim \mathcal{N}(\mu(X_*), K(X_*)) \tag{2}$$

*The kernel must be chosen carefully* as there exists a clear link between kernel functions and predictions [18], which contribute to the hyper-parameter optimization. From our experimental data, we observe *the influencing parameters on total energy consumption are sparse and vary over a broad range including both numerical and categorical variables*. Hence, we choose the automatic relevance determination (ARD) exponential squared kernel for our predictive model, which automatically puts different weights on the parameters with differential scales assessing their significance to the model. Hence, our kernel equation becomes:

$$K(x_i, x_j) = \sigma_f^2 \exp[(-\frac{1}{2}) \sum_{m=1}^{D} \frac{(x_{im} - x_{jm})^2}{\sigma_m^2}], \tag{3}$$

where $\sigma_f^2$ is the hyper-parameter to be optimized and $\sigma_m^2$ is the covariance of the $m^{th}$ dimension. Finally, the log-likelihood of the trained model can be expressed as

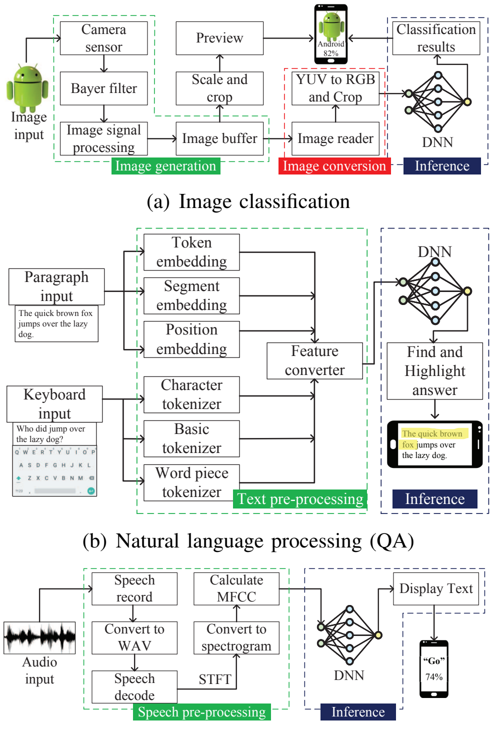
$$\begin{aligned} \log P(E(X)|X^{1:D}) = &-\frac{1}{2} E(X)^T (K + \sigma_D^2 I)^{-1} E(X) \\ &- \frac{1}{2} \log det(K + \sigma_D^2 I) - \frac{D}{2} \log 2\pi, \end{aligned} \tag{4}$$

where $I$ is an identity matrix. EPAM is first trained offline with the observation data points, then is run with an application alongside. The prediction is done either simultaneously or at the end of an application. In this research, we train the model with a dataset containing $85,500$ data, validate with $19,496$, and test with $10,000$ data.

## IV. Experimental Setup

**a) AI applications:** Three mobile AI applications are used in this research: image classification, NLP, and speech recognition. In image classification, as shown in Fig. 2(a), first, the image is captured by the camera sensor, which then goes through a Bayer filter and image signal processor, and, then is stored in an image buffer. The image frame is then scaled and cropped to be previewed while simultaneously going to an image reader, converted from YUV color format to RGB, and cropped according to the input size of the DNN model. Then the converted and cropped frame is taken as the DNN input, generating the classification results to display.

The NLP question-answer application takes both the paragraph input and the question input from the keyboard (Fig. 2(b). The paragraph is then represented with token, segment, and position embeddings. The keyboard input goes through character, basic, and word piece tokenizer. These embeddings and tokens are passed to a feature converter providing input to the DNN model. The model finds the answer to the question input and highlights it in the paragraph.



(a) Image classification

(b) Natural language processing (QA)

(c) Speech recognition

Fig. 2. Pipelines of the mobile AI applications studied in this research.

Speech recognition application records, converts, and decodes the audio input. The decoded audio signal is converted to a spectrogram by running a short-time Fourier transform (STFT) along with the calculation of the Mel frequency cepstral coefficients (MFCCs). The spectrogram and MFCC are passed to the DNN model. The predicted word is then displayed on the phone as depicted in Fig. 2(c).

**b) Testbed:** We implement the applications mentioned above on four Android OS-based smartphones from different manufacturers with distinct configurations to make the measurement study robust with a wide range of parameters. Table I shows the specifications of the smartphones used in the experiment. However, the intended thorough investigation of mobile AI brings several challenges during the experiment.

Android Studio, along with other third-party contributors, provides developers with memory and battery profilers, which cannot generate the data necessary to measure memory usage and power consumption precisely. In this experiment, we collect latency timestamp data of each segment of a mobile AI pipeline along with their corresponding memory usage. To measure the energy consumption, we use an external power measurement tool "Monsoon Power Monitor" that provides data sampled at every 0.2 ms interval. However, due to the delicate design of power input terminals, the latest smartphones need to be heated and opened to remove the battery, and then are connected to the power monitor. After careful measurement of power data, they are matched with the corresponding latency timestamps.

TABLE I
BRIEF SPECIFICATIONS OF THE DEVICES USED IN THE EXPERIMENTS

| Denotation | Model | SoC | CPU | GPU | Dedicated AI accelerator | RAM | OS | NNAPI support | Release Date |
|---|---|---|---|---|---|---|---|---|---|
| Device-1 | Huawei Mate 40 Pro | Kirin 9000 (5 nm) | 8-core (1x3.13GHz A77 3x2.54GHz A77 4x2.05GHz A55) | Mali G78 | Ascend Lite+ Tiny NPU Da Vinci 2.0 | 8GB LPDDR5 | Android 10 | Yes | October, 2020 |
| Device-2 | OnePlus 8 Pro | Snapdragon 865 (7 nm) | 8-core (1x2.84GHz 3x2.42GHz 4x1.8GHz Kryo 585) | Adreno 650 | Hexagon 698 DSP | 8GB LPDDR5 | Android 10 | Yes | April, 2020 |
| Device-3 | Motorola One Macro | Helio P70 (12 nm) | 8-core (4x2.0GHz A73 4x2.0GHz A53) | Mali G72 | MediaTek APU | 4GB LPDDR4X | Android 9 | Yes | October, 2019 |
| Device-4 | Xiaomi Redmi Note8 | Snapdragon 665 (11 nm) | 8-core (4x2GHz Gold 4x1.8GHz Silver Kryo260) | Adreno 610 | Hexagon 686 DSP | 4GB LPDDR4X | Android 10 | Yes | August, 2020 |

TABLE II
DNN MODELS USED IN THIS RESEARCH

| Denotation | Model Name | Application | Input size | No. of layers | Model Size |
|---|---|---|---|---|---|
| Model 1 | MobileNetV1 (Float) | Image classification | 224x224x3 | 31 | 16.9 MB |
| Model 2 | MobileNetV1 (Quantized) | Image classification | 224x224x3 | 31 | 4.3 MB |
| Model 3 | EfficientNet-lite (Float) | Image classification | 224x224x3 | 62 | 18.6 MB |
| Model 4 | EfficientNet-lite (Quantized) | Image classification | 224x224x3 | 65 | 5.4 MB |
| Model 5 | NASNet Mobile (Float) | Image classification | 224x224x3 | 663 | 21.4 MB |
| Model 6 | Mobile BERT QA | Natural language processing | int32 [1, 384] | 2541 | 100.7 MB |
| Model 7 | Tensorflow ASR | Speech recognition | [20 Hz, 4 kHz] | 8 | 3.8 MB |

To make the experiment environment controllable, we carry out all the experiments in a similar condition, e.g., brightness, camera focus, image resolution, background applications, processing sources, and test dataset. We use $640 \times 480$ pixels as the image resolution, and `TensorFlow Lite Delegate` to control the processing sources. The 2017 COCO test dataset, WH-questions, and fixed single words are used for testing the classification, NLP, and speech recognition, respectively. In addition, even without any applications running in the background, there is always a minimal power consumption – which we call the *base power*. To distinguish the mobile AI power from the base power, an additional layer is used before the actual AI application.

**c) AI models:** In this research, we use seven DNN models for three different applications. In Table II, the details of each model, including the input size, number of layers, and the trained model size (occupied storage space) are shown.

**d) Performance metric:** We evaluate all the AI applications' performances in terms of their latency, energy consumption, and memory usage. The total energy consumption is controlled by latency and memory usage, as well as the category of AI applications, processing sources, model types (float and quantized), and DNN structure and model size.

## V. RESULTS AND DISCUSSION

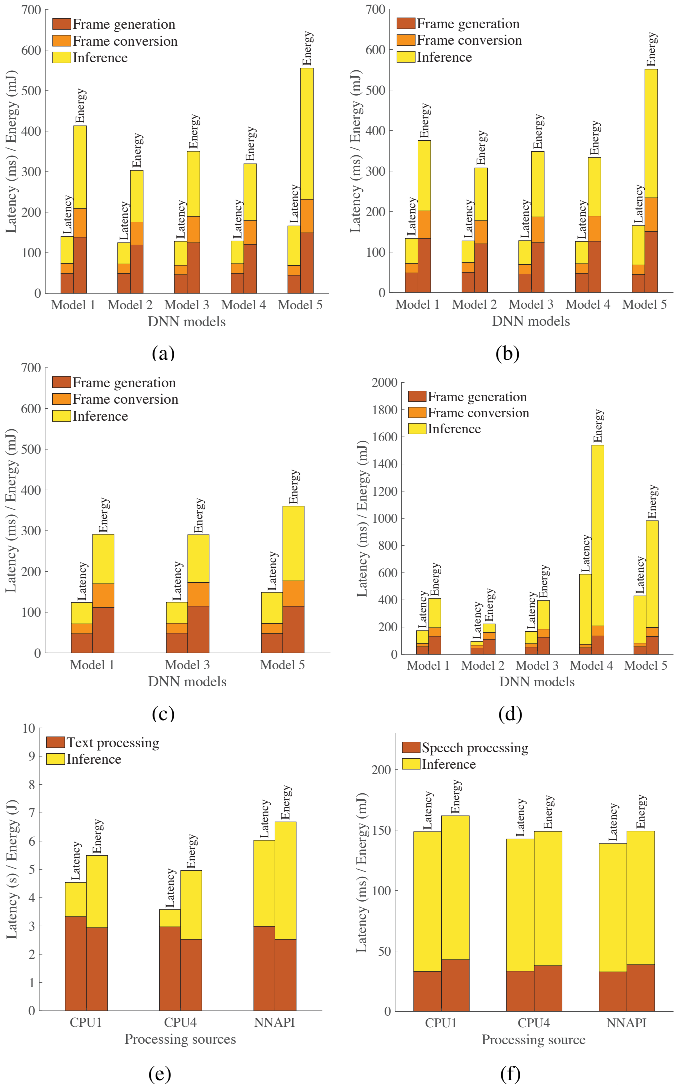
We conduct experiments with all the devices listed in Table I and models listed in Table II by switching to different processing sources, such as CPU thread 1 and thread 4, GPU, and NNAPI. Models 1 to 5 are for vision-based AI, and models 6 and 7 are for non-vision-based AI applications. It is to be noted that models 2, 4, 6, and 7 do not support GPU processing due to a lack of `TensorFlow Lite` optimization. In general, the applications have input data processing (combining image generation and conversion in classification) and inference tasks. In this paper, we show some of the interesting findings due to space constraints.

### A. Latency and energy consumption of mobile AI

The end-to-end latency and energy consumption per cycle for all the models with different processing sources are shown in Fig. 3. First, we can see that quantized models decrease the inference latency (13%) and energy consumption (25%) from their respective float models. Additionally, there is a reduction in the overall latency of 4% when switching to a 4-thread from a single-thread CPU. However, in quantized models, the multi-thread CPU processing slightly increases the total energy consumption (3% on average). The use of GPU even lowers the end-to-end latency and energy consumption compared to the use of single-thread CPU (8% and 27% respectively on average) and 4-thread CPU (7% and 25% respectively on average). On the contrary, NNAPI behaves differently than the other three processing sources on different devices. For models 4 and 5, NNAPI increases latency and energy considerably. Our insight here is that NNAPI can perform better with sufficient hardware support from the manufacturers.

An interesting fact about the NLP application is that the text processing step shows an entirely different latency pattern. This segment takes user input which does not take uniform time, i.e., it varies with user habits of typing and thinking of the question. Hence, the processing stage here is completely unpredictable for different users. In NLP, each input consumes around 5.7 J, whereas, another non-vision application, speech recognition takes around 161.85 mJ to process one speech input sampled at a rate of 16 kHz using a single-thread CPU. NNAPI consumes the least latency and energy for speech recognition.

In addition, we examine the power consumption charts of different applications and processing sources (Fig. 4). We observe a slight initiation delay for every application (marked with red arrows in Fig. 4), which varies with using different processors and applications. This delay occurs during the time when the application interface initiates till the activity-start point, which is mainly originated by different hardware components being accessed at the beginning of an AI application, such as the



(a)

(b)

(c)

(d)

(e)

(f)

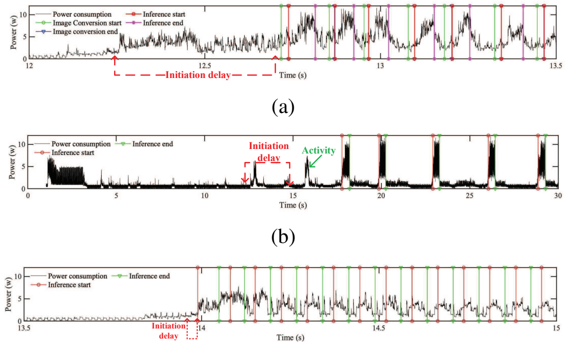
Fig. 3. End-to-end mean latency and energy consumption per cycle of vision-based models 1–5 for (a) single- and (b) multi-thread CPU, (c) GPU, and (d) NNAPI, and non-vision-based (e) model 6 and (f) model 7.

camera, keyboard, speaker, and microphone. Besides, different processor delegations (e.g., GPU and NNAPI) are also done during this period.

**Highlights:** *Non-vision applications cannot be generalized for latency and energy like vision-based ones. GPU processing is not supported by non-vision applications, which should be explored widely. The initiation delay (i.e., the delay between the activity trigger and start point) varies along AI models, processing sources, and applications, which is caused by accessing different hardware components by mobile AI applications.*

## *B. DNN structures and their inference latency and energy*

DNN structures define the way inference activities work in a mobile AI application. The behavior of DNN structures varies across different kinds of applications as well, e.g., vision and non-vision AI. For instance, a smaller DNN structure for vision applications can incur higher latency and energy than a larger non-vision DNN structure. Inference latency and energy consumption per cycle are shown in Fig. 5 for DNN models with single-thread CPU processing. We observe that model 5 takes longer inference time and energy due to its larger structure than the other vision-based AI models. The longest latency and highest energy are evident in model 6 (a complex structure comprising 2541 layers).



(a)

(b)

(c)

Fig. 4. Power consumption pattern for (a) classification, (b) NLP, and (c) speech recognition.

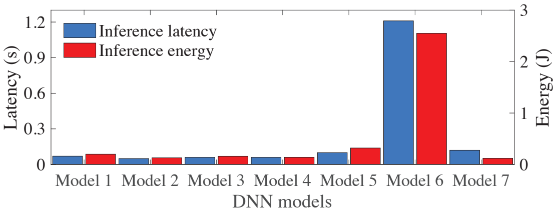


Fig. 5. Inference latency and energy consumption per cycle by DNN models.

**Highlights:** *DNN structures influence inference latency and energy significantly, but the relationship is not linear at all. Generally, larger DNN structures are responsible for higher latency and energy for a mobile AI application.*

## *C. DNN model size, memory usage, and inference energy*

DNN model size (i.e., the storage space occupied by the model) impacts memory usage and energy consumption during inference. From our experiment, we observe that model 7 has the lowest model size, hence causing the lowest memory and energy consumption, whereas model 6 has the highest size, memory, and energy consumption. This is more evident from Fig. 6, which shows a comparison among all the models' sizes, inference memory, and energy consumption for single-thread CPU processing.

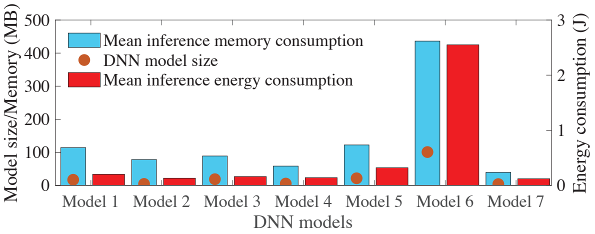
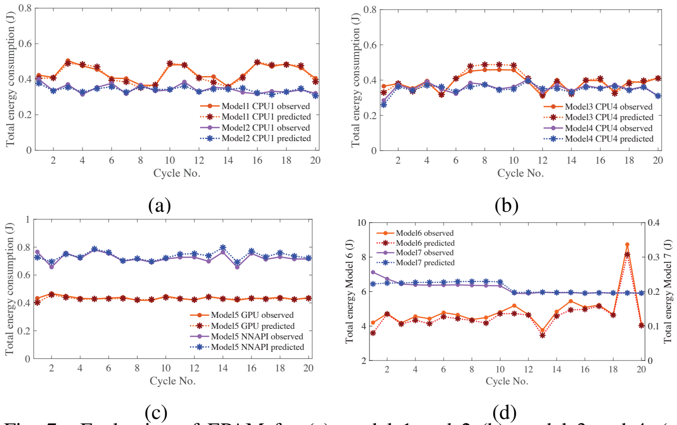


Fig. 6. Comparison of DNN model size, inference memory usage, and inference energy consumption.

**Highlights:** *Lower memory used by mobile AI applications ensures computation resources and energy for other mobile device activities. From this perspective, quantized and smaller DNN models are best suited for mobile AI. The larger the storage occupied by a DNN model, the higher the memory and energy consumption.*

### D. Performance evaluation of EPAM

We develop and train the Gaussian process regression-based predictive energy model, EPAM, with each device's SoC, CPU frequency, no. of cores, memory size, processing sources, no. of threads, application type, DNN model, DNN structure, memory usage, processing latency, and inference latency from the large experimental dataset from this research to predict the total energy consumption per application cycle (data processing and inference for each input). We use an empty basis function, and ARD squared exponential kernel function for the hyper-parameter optimization. We use device-1, 2, and 4 for training and validation, and device-3 for 1-step ahead prediction testing. Due to page limitation, we show only a few prediction results in Fig. 7. We observe that EPAM's energy prediction per cycle is highly accurate for all the models. The overall root mean squared error (RMSE) is 0.075 (3.06%), and the marginal log-likelihood value is $-1.449 \times 10^2$, which show that the trained model is a good fit for the prediction. The prediction latency depends on the machine used in running the model.



(a)

(b)

(c)

(d)

Fig. 7. Evaluation of EPAM for (a) model 1 and 2 (b) model 3 and 4, (c) model 5, and (d) model 6 and 7 with different processing sources.

**Highlights:** *EPAM further helps developers and users to perceive the performance of individual AI applications in terms of energy with high accuracy – which is the primary motivation of this research work. The larger and more diverse the training dataset, the higher the prediction accuracy.*

## VI. Conclusion

In this paper, we presented a comprehensive study of mobile AI applications with different processing sources and AI models. Overcoming the challenges with measurement, we conducted experiments to assess the performance of different AI models, processing sources, and devices. Our measurement work shows that the latency, energy consumption, and memory usage vary based on DNN models and processing sources. Mobile AI systems' performance is substantially improved using quantized models than floating-point models in terms of latency and energy. Another important finding is that the storage space occupied by DNN models influences the memory and energy consumed during inference almost linearly. Additionally, non-vision applications follow a different trend of latency and energy consumption than vision-based AI since their input processing techniques differ from vision applications. Every AI application has an initiation delay caused by accessing various hardware components of mobile devices, which varies for different models and configurations. Moreover, the latency, memory, AI model, and device configuration impact the total energy consumption for a complete application cycle, albeit at different correlations. This non-linear correlation in a non-parametric model led to our proposed predictive energy model, EPAM, based on Gaussian process regression. Finally, we trained and validated EPAM with the vast dataset obtained from our experiment. The evaluation of EPAM shows high accuracy with an overall RMSE of 0.075 (3.06%). Developers can use EPAM to predict the energy consumption of their mobile AI applications without measuring the energy externally to improve the comprehensive user experience. To summarize, this novel predictive energy model, EPAM, will help the mobile AI research community design energy-improved applications considering all the control factors and parameters that can reduce energy requirements to enable better service for smartphones, wearable devices, and autonomous vehicles.

## References

[1] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proc. of IEEE CVPR*, 2016.

[2] L. N. Huynh, R. K. Balan, and Y. Lee, "DeepSense: A GPU-based deep convolutional neural network framework on commodity mobile devices," in *Proc. of ACM Workshop on Wearable Systems and Applications*, 2016.

[3] M.-Y. Lai, C.-Y. Sung, J.-K. Lee, and M.-Y. Hung, "Enabling Android NNAPI flow for TVM runtime," in *Proc. of ACM ICPP: Workshops*, 2020, pp. 1–8.

[4] C. Luo, X. He, J. Zhan, L. Wang, W. Gao, and J. Dai, "Comparison and benchmarking of AI models and frameworks on mobile devices," *arXiv preprint arXiv:2005.05085*, 2020.

[5] H. Wang, B. Kim, J. Xie, and Z. Han, "Energy drain of the object detection processing pipeline for mobile devices: Analysis and implications," *IEEE Trans. on Green Communications and Networking*, vol. 5, no. 1, pp. 41–60, 2021.

[6] A. Mallik and J. Xie, "H.264 video encoding-based edge-assisted mobile AR systems: Network and energy issues," in *Proc. of IEEE ICC*, 2022, pp. 1046–1051.

[7] A. Ignatov, R. Timofte, A. Kulik, S. Yang, K. Wang, F. Baum, M. Wu, L. Xu, and L. Van Gool, "AI benchmark: All about deep learning on smartphones in 2019," in *Proc. of IEEE/CVF Workshop*, 2019.

[8] J. Cheng, J. Wu, C. Leng, Y. Wang, and Q. Hu, "Quantized CNN: A unified approach to accelerate and compress convolutional networks," *IEEE Trans. on Neural Networks and Learning Systems*, 2017.

[9] F. N. Iandola, A. E. Shaw, R. Krishna, and K. W. Keutzer, "SqueezeBERT: What can computer vision teach NLP about efficient neural networks?" in *Proc. of Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, 2020.

[10] J. Liu, J. Liu, W. Du, and D. Li, "Performance analysis and characterization of training deep learning models on mobile device," in *Proc. of IEEE ICPADS*, 2019, pp. 506–515.

[11] D. Shin, J. Lee, J. Lee, J. Lee, and H.-J. Yoo, "DNPU: An energy-efficient deep-learning processor with heterogeneous multi-core architecture," *IEEE Micro*, vol. 38, no. 5, pp. 85–93, 2018.

[12] A. Pathak, Y. C. Hu, and M. Zhang, "Where is the energy spent inside my app? Fine grained energy accounting on smartphones with eprof," in *Proc. of ACM European Conference on Computer Systems*, 2012.

[13] A. Jindal and Y. C. Hu, "Experience: Developing a usable battery drain testing and diagnostic tool for the mobile industry," in *Proc. of ACM MobiCom*, 2021, pp. 804–815.

[14] H. Wang, B. Kim, J. Xie, and Z. Han, "LEAF+AIO: Edge-assisted energy-aware object detection for mobile augmented reality," *IEEE Trans. on Mobile Computing*, 2022.

[15] Q. Cao, A. Balasubramanian, and N. Balasubramanian, "Towards accurate and reliable energy measurement of NLP models," in *Proc. of Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, 2020.

[16] T. Ullah, A. H. Siraj, U. M. Andrabi, and A. Nazarov, "Approximating and predicting energy consumption of portable devices," in *Proc. of IEEE International Conference on Information Technology and Nanotechnology (ITNT)*, 2022, pp. 1–7.

[17] J. Wang, "An intuitive tutorial to Gaussian processes regression," *arXiv preprint arXiv:2009.10862*, 2020.

[18] M. Seeger, "Gaussian processes for machine learning," *International Journal of Neural Systems*, vol. 14, no. 02, pp. 69–106, 2004.