Wildlife Species Classification on the Edge: A Deep Learning Perspective

Subodh Ingaleshwar[1][a], Farid Thasharofi[1][b], Mateo Avila Pava[1][c], Harshit Vaishya[1][d], Yazan Tabak[1][e], Juergen Ernst[1][f], Ruben Portas[2][g], Wanja Rast[2][h], Joerg Melzheimer[2][i], Ortwin Aschenborn[2][j], Theresa Goetz[1,3][k] and Stephan Goeb[1][l]

[1]*Fraunhofer- Institute for Integrated Circuits IIS, Erlangen, Germany*
[2]*Leibniz Institute for Zoo and Wildlife Research, Berlin, Germany*
[3]*Department of Industrial Engineering and Health, University of Applied Sciences Amberg-Weiden, Germany*

Keywords: Artificial Intelligence (AI), Animal Species, Applied Conservation, Deep Neural Networks, Embedded Systems, Energy Efficient, Image Classification.

Abstract: Accurate and timely recognition of wild animal species is very important for various management processes in nature conservation. In this article, we propose an energy-efficient way of classifying animal species in real-time. Specifically, we present an image classification system on a low power Edge-AI device, which embeds a deep neural network (DNN) in a microcontroller that accurately recognizes different animal species. We evaluate the performance of the proposed system using a real-world dataset collected via a small handheld camera from remote conservation regions of Africa. We implement different DNN models and deploy them on the embedded device to perform real-time classification of animal species. The experimental results show that the proposed animal species classification system is able to obtain a remarkable accuracy of 84.30% with an energy efficiency of 0.885 mJ on an edge device. This work provides a new perspective toward low power, energy-efficient, fast and accurate edge-AI technology to help in inhibiting wildlife-human conflicts.

1 INTRODUCTION

The illegal trade in wildlife products is a global problem. This is not only endangering animal species that are already at risk of extinction but also affecting the livelihoods and security of human lives residing in the region (Wildlife Crime Report, 2022). It is a recorded fact that in every 20 minutes, an animal is poached or killed in human-wildlife conflict (Poaching and Biodiversity Report, 2022). According to World Wildlife Fund (WWF) for Nature, poaching of Cheetahs has increased to 7,700% in last few years (WWF Report, 2021). Zoologists are of the opinion, the more we study the wild, better we can develop and apply effective conservation measures. Artificial Intelligence (AI) on edge devices is expanding to more niche domains, for instance ecological understanding, because of the wide range of advancement in the areas of embedded systems design (Dominguez et.al., 2021) (J. Bartels et.al., 2022). Areas of embedded systems design include high-speed parallel processing elements, ultra-lower boards, multi-level PCB design and IDE's with low-level debug features. These advancements, from AI model development to deployment, lead to a new set of tools and processes in DNN powered embedded AI applications.

[a] https://orcid.org/0000-0002-4425-1317
[b] https://orcid.org/0009-0006-1822-7889
[c] https://orcid.org/0009-0008-3061-8588
[d] https://orcid.org/0009-0007-8150-8576
[e] https://orcid.org/0009-0000-4981-9090
[f] https://orcid.org/0009-0007-6600-2745
[g] https://orcid.org/0000-0002-0686-0701
[h] https://orcid.org/0000-0003-3465-3117
[i] https://orcid.org/0000-0002-3490-1515
[j] https://orcid.org/0000-0002-7494-3795
[k] https://orcid.org/0000-0001-8751-3404
[l] https://orcid.org/0000-0002-1206-7478

Ingaleshwar, S., Thasharofi, F., Pava, M., Vaishya, H., Tabak, Y., Ernst, J., Portas, R., Rast, W., Melzheimer, J., Aschenborn, O., Goetz, T. and Goeb, S.
Wildlife Species Classification on the Edge: A Deep Learning Perspective.
DOI: 10.5220/0012376700003636

In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 3*, pages 600-608
ISBN: 978-989-758-680-4; ISSN: 2184-433X

1.1 Challenges in AI

As the magnitude of the features in Neural Network's (NN) crosses one billion trainable parameters, increment in storage & arithmetic operations prevents them from being adopted in the battery powered embedded environments. Embedded AI and Edge AI are AI technologies related to the deployment of AI models on Local/Edge devices, rather than relying on centralized cloud-based solutions. However, they have different focuses and use cases. Edge devices are the devices with limited power capacity like smartphones, smart sensors, wearables etc. Edge devices are preferred over the cloud for certain applications due to data privacy, less latency, limited battery power and limited communication bandwidth (Edge AI Technology Report, 2023). The main difference between embedded AI and edge AI is the scale and complexity of AI tasks that can be handled and the types of devices to be deployed. Embedded AI refers to specific functions within dedicated hardware, whereas Edge AI is more versatile and can be deployed on a broader range of devices for real-time, local processing. The choice between them depends on the specific use case and availability of hardware resources.

1.2 Related Work

Several studies discuss the different deep learning based methods for classifying different animal species. Authors (S. Han et.al., 2021) proposed four different methods of animal species classification using Face HQ dataset. Two convolutional neural network (CNN) based VGG16 and ResNet methods achieved an accuracy of 84% and 87% respectively. The remaining two unsupervised clustering with variational auto-encoder and auto-encoder with SVM records almost 94% of accuracy over the test data. The proposed methods recorded good accuracy but contain complex computations that make it hard to synthesize. Authors (Sahil Faizal et.al., 2022) provided a method for classifying animals mentioned in IUCN Red List of Threatened Species. They proposed a technique based on fine-tuning of the InceptionResNet that has been trained using cloud computing resources of Google Colab on animal species from Kaggle dataset. The recorded test accuracy is 95% with less number of epochs. The network performs complex computations, which are difficult to synthesize. Authors (Binta Islam, S. et.al., 2023) proposed an AI-based automated classification solution for camera-trap, herpetofaunal animals using the pre-trained DNN models like ResNet and VGG16. Authors (Zualkernan, I et.al.,2022) introduced an IoT system for animal species classification using pre-trained models like InceptionV3, MobileNetV2, ResNet18, EfficientNetB1, DenseNet121, and Xception neural network models. They used a custom made camera-trap image dataset of 66 thousands images and deployed on different platforms. The latency time for Jetson Nano is 0.276 sec with current consumption of 1665.21 mA and for Raspberry-pi is 838.99mA with latency of 2.83sec. (Zhongqi Miao et.al.,2019) suggested a DNN model using VGG16 and ResNet50 along with gradient weighted class-activation-mapping (Grad-CAM) procedure to extract the most salient features in the final convolution layer. The proposed method reported an accuracy of 86%.

Authors (Ibrahim Mai et.al., 2020) recorded accuracy of 96% on their CNN model, which is trained using BCMOTI and Snapshot Wisconsin datasets. The recorded an inference time is 9sec, which is high, compared to the other controllers (Arthur Moss et.al.,2022) (Mitchell Clay et.al.,2022) like MAX 78000 with inference time varies from 3 to 26ms based on model and input size. Authors (A. Reuther et.al., 2020 & 2022) provide an extensive list of accelerators categorized as very low power, autonomous, data center chips and cards, lastly data center systems. Authors also provide the sorted list based on different features like computation precision, form factor, peak performance and power consumption details. Similarly, author Weison Lin et.al, not only lists pre-configured edge AI accelerators but also coarse-grained reconfigurable array (CGRA) technology accelerators, which support dynamic reconfiguration (Lin, W. et al., 2021). Author also mentions actual performance, implementation, and productized examples of edge AI accelerators with key performance metrics that can be of significant information for Embedded AI designers.

This paper demonstrates an entire framework of the animal classification system starting from training of CNN based classification model to its deployment onto a low-power Edge device MAX 78000FTHR. The system is specifically built for deep learning based applications with an on-board CNN accelerator. The main contributions of this work are summarized as below,

a) Developing an end-to-end ultra-low powered image classification system for recognizing different animal species.
b) Perform a thorough experimental analysis of two different DNN models that efficiently classify different animal species on the edge device.

The remaining part of the paper is organized as follows: Section II describes the materials like datasets, components and methods used to build model and selection of the AI hardware. Section III presents and discusses the results obtained from the proposed system. The paper ends with a conclusion in Section IV.

2 MATERIAL AND METHODS

2.1 Dataset

The image data used in this study are collected using a range of available cameras including cell phone cameras over the vast and remote regions of Namibian savannah ecosystems. The dataset contains 6300 images of three different classes: Elephant, Cheetah and landscape. Duplicate or similar images were removed manually and only 5550 were considered for experimentation. The experiment dataset contains 1650 images of Elephants, 1650 images of Cheetahs and 1800 images of landscape, all of which were labeled and examined by the researchers of Leibniz-IZW. Figure 1 shows the examples of images from each class used in our study.

Figure 1: Example of the dataset. The wild animals are shown in the top row (Elephant and Cheetah, respectively), while landscape class for this experiment are displayed in the bottom row.

2.2 Image Pre-Processing

The collected images had a resolution of 5472×3648 pixels. All the images were down-sampled to a resolution of 64×64 , 96×96 and 180×180 pixels. In order to increase the model's generalizability, data augmentation techniques were applied. Random transformations such as horizontal flip, rotation (90 degree), Gaussian blur and Color augmentation were performed on each image. The dataset is split into 90% training images and 10% testing images. The test data is treated as unseen data and only reserved for testing the model. We perform 5-fold cross-validation on the training dataset to fine-tune each of the selected models. After achieving satisfactory accuracy through the cross-validation processes, the model with best performance was selected as the final model and evaluated with unseen test data.

2.3 Deep Neural Network (DNN) Models

In this study, two DNN models were trained and tested on the collected dataset. The selected DNN models are inspired from popular VGG architecture (Karen Simonyan and Andrew Zisserman, 2015) with varying number of convolutional layers such as, six layer VGG (VGG-6) and eight layer VGG (VGG-8). VGG-6 consists of three convolutional layers and three fully connected layers, whereas VGG-8 is made up of five convolutional layers and three fully connected layers. The last fully connected layer is a softmax layer with three neurons for predicting the corresponding classes. Both the models were trained using Adam optimizer and cross-entropy loss function. Furthermore, both the models were trained using the PyTorch framework and then retrained with MAXIM API's.

2.4 Hardware/Deployment Platform

To evaluate the performance of the selected DNN models in real-time, we deployed them onto the ultra-low power edge device. We have analyzed seven different recently launched ultra-low powered embedded processors mainly used for neural network inference and training purposes. These accelerators are Maxim 78000 (Maxim User Guide, 2020), GAP8, GAP9 (GAP Processors, 2020), Kendryte (L.Gwennan, 2019), Perceive (J.McGregor, 2020), AI storm, Gyrfalcon (SolidRun, 2020). All these AI accelerators are on-chip devices intended for low powered-low latency applications.

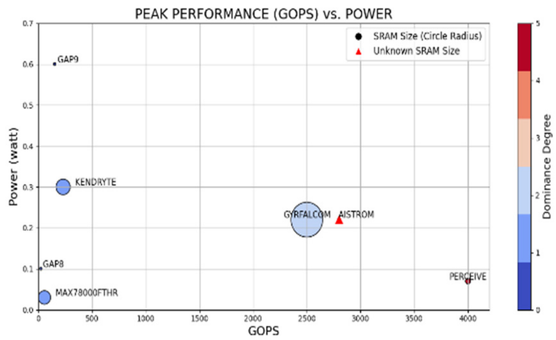

Figure 2: Pareto diagram for AI hardware selection.

Figure 2 shows the Pareto diagram used for the selection of an efficient hardware model. The key factors considered in this graph are peak performance in terms of Giga operations per sec (GOPs/sec), Peak power (w) and SRAM size of each AI processor. Circle with varying radius is used to denote the SRAM size. Larger the radius, higher the size of SRAM and vice versa. After studying the Pareto diagram shown in Figure 2, we choose the AI accelerator with the best performance such as MAX78000 CNN inference engine in this study.

Table 1: Main features of MAX78000.

Main features	MAX78000
ARM Cortex M4 with FPU	Operating @100MHz
NN Accelerator with 64 parallel processors	Operating @ 50MHz
RISC V as Smart DMA	Operating @ 60MHz
Operating modes of MAXIM 78000	Seven operating modes (Active, Sleep. Low power mode, Ultra-low Power, Standby, Backup, Power down)
Flash memory	512KB
SRAM	128KB
NN Accelerator RAM	Data RAM 512KB Mask RAM 432KB Bias RAM 2KB Tornado RAM 384KB

The MAX78000FTHR is a new Artificial Intelligence (AI) board with MAX78000 microcontroller that enables DNN models to operate in real-time at ultra-low power (Maxim User Guide, 2020). This controller has an ARM Cortex-M4F core, a RISC-V core, and a CNN accelerator, which enables low-powered applications to run AI inferences at high speed while consuming very low energy. The main features of the MAX78000 are summarized in Table 1. The selected VGG-6 and VGG-8 models are deployed onto the MAX78000FTHR using PyTorch checkpoints. Since the PyTorch models are trained with floating-point weights and biases, weights are quantized using integer-arithmetic-only quantization during re-training with MAXIM API's. The model's performance is expected to be degraded due to weight quantization. The quantized model is synthesized using MAX78000 synthesizer via Maxim tools (Maxim User Guide, 2020). The C code generated from the MAX78000 synthesizer is then executed on the MAX78000 to predict the class of unseen images in real-time.

2.5 Maxim Micros SDK: Firmware Development Using MaximSDK

The Maxim Micros SDK (Maxim User Guide, 2020) is a multi-os installer used to install the Eclipse IDE, examples, libraries and necessary tools required to develop the firmware for Maxim Integrated's Microcontroller ICs. This installer is fully integrated with Eclipse™ and MaximSDK. The Eclipse IDE is used for C/C++ project development, with peripheral configuration, code generation, code compilation and low level debug features for MAXIM micro-controllers. It also bundles setups for all the required programs. The programs bundled in the setup consist of GNU Tools for ARM Embedded Processors, Eclipse CDT IDE for C/C++ Developers (Maxim Integrated version), Maxim Integrated Bitmap Converter, Maxim Integrated Secure Tools, Minimalist GNU for Windows (MinGW), Open On-Chip Debugger(OpenOCD), and Olimex ARM-USB-TINY-H Drivers.

3 RESULT ANALYSIS

In this section, we present the experimental results obtained from the classification system using the PyTorch framework, over two different DNN models and different image resolutions. These experiments were performed in two different testing scenarios: (1) training and testing the DNN models using PyTorch framework on a dedicated computer, and (2) deploying the trained DNN models on MAX78000 using Maxim development tool and testing unseen images in real-time. We further provide a thorough analysis of a real-time image classification approach that significantly influences the testing accuracy, inference time, memory utilization and energy consumption when deployed onto the edge device.

3.1 Model Evaluation Results

The results obtained from both DNN models for the collected dataset after training and evaluating them on unseen test data using PyTorch are presented in Table 2. A model's performance can be assessed by how the trained classifier predicts the unseen image. First, to assess the effect of image resolution on validation accuracy, input images are down-sampled to three different sizes. These sizes include dimensions of 64×64, 96×96, and 180×180. This analysis helps in understanding how down-sampling affects the DNN model's validation accuracy. Figure 3 demonstrates the validation accuracy of the VGG-8 model over 100 training epochs. It is observed that the model has been converged around 40 epochs. From Figure 3, it can be observed that down-sampling input images to a resolution of 64×64 leads to performance degradation, compared to using images with a resolution of 180×180.

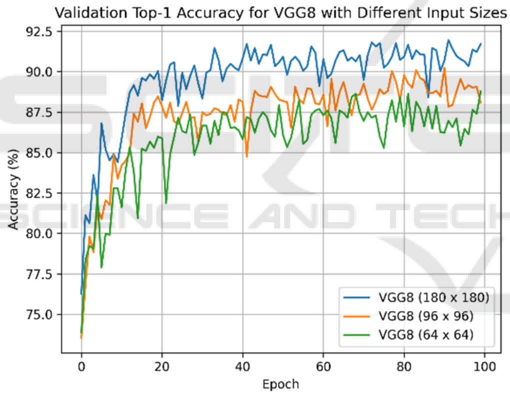

Figure 3: Validation accuracy for VGG-8 model to investigate the effect of down-sampling on performance during 100 training epochs.

We then evaluated the model's performance with a test dataset to measure the model's generalizability. The test results were calculated with commonly used statistical metrics known as accuracy and F1 score. The results obtained over the different image resolutions for each of the DNN model configurations are reported into Table 2. As expected, higher image resolution (180×180) achieved higher accuracy of 84.45% and 88.12% for VGG-6 and VGG-8 models, respectively. A higher resolution implies the availability of more information in terms of more pixels to classify the image. On the other hand, when images were down-sampled to the dimensions of 64×64, the classification performance of both models degraded significantly. However, in order to achieve the best classification performance, the higher image resolution to be used which directly affects the energy consumption of the edge device on which models are deployed, as well as the total inference time to perform prediction for each image.

Table 2: Classification results of VGG-6 and VGG-8 with unseen test dataset.

Model	Image Size	Accuracy [%]	F1 score [%]
VGG-6	64x64	82.88	80.05
	96x96	83.12	82.4
	180 x 180	84.45	82.67
VGG-8	64x64	81.55	79.67
	96x96	86.67	85.93
	180 x 180	88.12	86.53

3.2 Embedded AI Deployment Results

The already-trained DNN models obtained with PyTorch were then quantized and synthesized using the Maxim tool in order to integrate them onto the hardware platform, MAX78000. Motivated by (Dominguez et.al., 2021), we used X-accuracy as a performance metric to demonstrate the performance of DNN models on MAX78000. It represents the difference in terms of accuracy with a model trained in PyTorch framework before quantizing and synthesizing it and after deploying it on MAX78000.

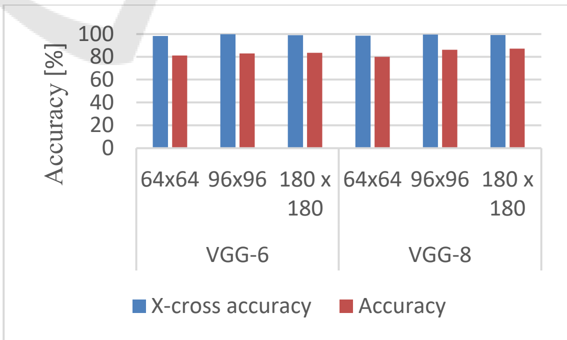

Figure 4: X-cross accuracy [%] and Accuracy [%]. of VGG-6 and VGG-8 with unseen test dataset when deployed on MAX78000. X-cross accuracy of 100% indicates that the model deployed on MAX78000 observed the same accuracy as the original model.

Therefore, 100% X-cross accuracy means the model deployed on MAX78000 obtains same accuracy as the model before deployment. The classification performance of each DNN model configuration deployed on MAX78000 is reported in Figure 4. Figure 4 shows that both VGG-6 and VGG-8 models with different image resolutions achieve almost the same accuracy as their software counterparts. The bars in red color indicate the actual accuracies obtained by each model when deployed on MAX78000.

3.2.1 Inference Time

Figure 5 presents the effect of image resolution and model size (in terms of number of convolutional layers) on inference time when predicting a single image on MAX78000. Figure 5 clearly shows that the smaller size images provide faster inference when deployed onto the edge device. As expected, the deeper the model, the more time it needs for prediction of an unseen image. For instance, the VGG-8 model clearly requires more time to perform prediction than that of VGG-6. Since the CNN accelerator of MAX78000 has 64 processors and a maximum 64 number of operations can be performed in parallel, the inference time is shown to be increased in a stepwise manner with different image resolutions. This is a quite interesting fact about MAX78000.

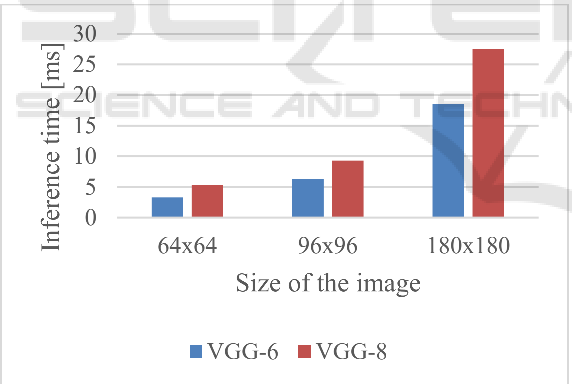

Figure 5: Inference time by each studied DNN model to perform a prediction for a single image on MAX78000.

3.2.2 Mops/S per Watt for Each of the DNN Models on MAX78000

Figure 6 indicates the performance of each studied model in terms of Mops/s/Watt (10^6 operations per second per watt) when deployed on MAX78000. This measure is the most commonly used to depict the performance of an embedded platform. As expected, a deeper model needs more operations to execute per demanded watt. Moreover, bigger images require a large number of operations to execute per demanded watt. On the other hand, a smaller image size and less deep model seem to be more efficient. An image with higher resolution entails the device must analyze more pixels in order to perform a prediction.

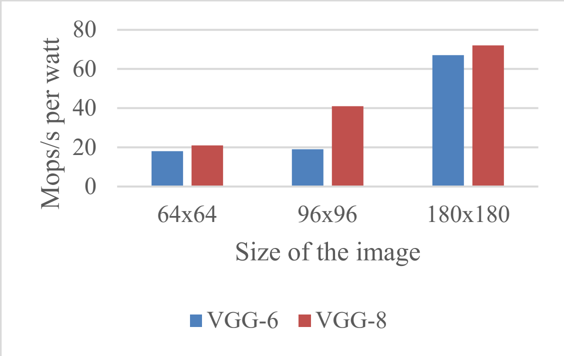

Figure 6: Mops/s per watt for each of the studied DNN models.

3.2.3 Memory Usage

Figure 7 compares the memory usage for both VGG-6 and VGG-8 in terms of flash, weight and bias memories. The reported results show that, in comparison to the higher resolution, smaller size images reduce the total required memory utilization (or usage) during inference. Higher resolution means an increase in the number of pixels, which in turn increases the memory consumption and prediction latency. Increasing the memory usage and the prediction latency directly increases the energy consumption that can be confirmed in the following Subsection 3.2.4. Bias memory utilized by VGG-6 is 2 bytes, whereas VGG-8 utilizes 514 bytes of bias memory during inference. The bias memory usage is not displayed in Figure 7 because of the scale of Y-axis.

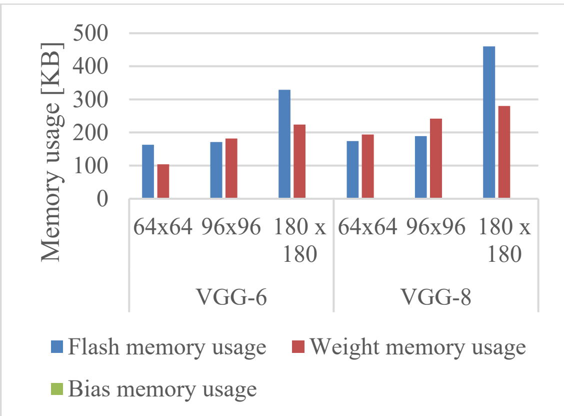

Figure 7: Memory usage for each of the studied DNN models on MAX78000.

3.2.4 Energy Consumption

Energy consumption is a crucial factor when deploying the DNN model onto an AI edge device. It indicates the capability of running DNN models using mJ's means of energy. We calculated the energy consumption of each model by measuring the current drawn while running each model. A simple setup used for the measurement of energy consumption is shown in Figure 8. Another crucial metric for measuring the energy consumption is inference execution time. Executing an inference on an AI device involves different operations such as setting it up, loading weights and data, executing the model, and offloading any result to the microcontroller unit. Figure 9 shows the current profile of VGG-8 model for an image size of 96×96. Total energy consumption is calculated by multiplying applied voltage, current drawn during inference, and inference execution time. Table 3 displays the execution time for each of the operations and current drawn during each operation.

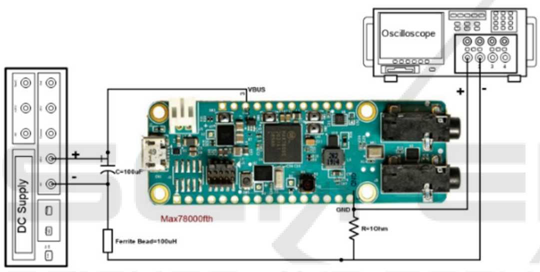

Figure 8: Experimental setup for inference energy measurement of AI device.

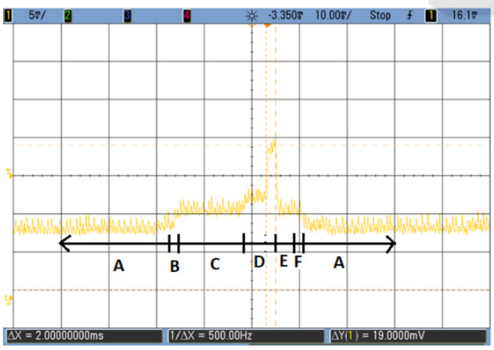

Figure 9: Current profile of the VGG-8 model for an image size of 96×96 (Note: The noise in the signal is due to onboard voltage regulator).

Figure 10 represents the energy consumed during an inference for each model on MAX78000. As expected VGG-6 with an image size of 64×64 consumed the least amount of energy. Overall, VGG-6 at all sizes consumes less energy than VGG-8 since it is a smaller model and needs fewer operations to execute on MAX78000. Higher image resolution based models consumed more energy, from 2.84 mJ to 4.275 mJ. This clearly indicates the influence of image resolution on energy consumption when the model is deployed on MAX78000.

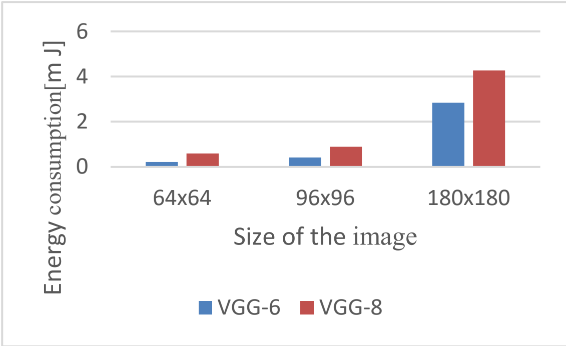

Figure 10: Energy consumption for each of the studied DNN models on MAX78000.

Table 3: Current drawn during and execution time of each operation to measure the energy consumption of VGG-8 model with an image size of 96×96.

Region	Operation	Current [mA]	Execution time [ms]
A	ARM Cortex – Active (CNN idle)	8	--
B	CNN enable	8 - 12	4
C	CNN configuration	12	14
D	Inference	19	9
E	Post Inference	12	5
F	CNN disable	8	3
Inference energy $E = V \times I \times Inference\ time$ $= 5V \times 19mA \times 9ms = 0.855\ mJ$			

4 CONCLUSION

In this paper, we performed the classification of animal species using an ultra-low power edge AI device named as MAX78000FTHR board. We have provided thorough analysis pertaining to animal species classification performance and real time performance implications for wildlife monitoring. We investigated the performance degradation exhibited when down-sampling input images, and demonstrated that significantly reducing the image resolution has a marginal effect on validation as well test accuracy, inference time, memory utilization and most importantly energy consumption. The experimental findings imply that the selected edge device, MAX78000 specific model optimization, need to be done to enhance the acceleration benefits. The AI device used here represents a suitable platform for future low power implementations in edge computing devices.

ACKNOWLEDGEMENTS

The authors would like to thank the Fraunhofer Institute for Integrated Circuits (IIS) for providing infrastructure for carrying out this research work and the European Research Consortium for Informatics and Mathematics (ERCIM) for the award of Research Fellowship.

REFERENCES

Wildlife Crime Report (2022): "Partnership against Wildlife Crime (in Africa and Asia)" published by Deutsche Gessellschat fur International Zusammenarbeit on behalf of Federal Ministry for Economic Cooperation and Development & Federal Ministry for the Environment, Nature and Nuclear Safety, https://www.bmz.de/resource/blob/100510/partnership-against-wildlife-crime.pdf

Poaching and Biodiversity Report: Combating poaching and the illegal trade in wildlife products

https://www.bmz.de/en/issues/biodiversity/poaching

World Wildlife Fund Report on Illegal Wildlife Trade: https://www.worldwildlife.org/threats/illegal-wildlife-trade

Dominguez-Morales J.P., Duran-Lopez L., Gutierrez-Galan D., Rios-Navarro A., Linares-Barranco A., Jimenez-Fernandez, A. (2021) "Wildlife Monitoring on the Edge: A Performance Evaluation of Embedded Neural Networks on Microcontrollers for Animal Behavior Classification". Sensors 2021, 21, 2975.

J. Bartels *et al*. (2022). "TinyCowNet: Memory- and Power-Minimized RNNs Implementable on Tiny Edge Devices for Lifelong Cow Behavior Distribution Estimation," in *IEEE Access*, vol. 10, pp. 32706-32727, 2022, doi: 10.1109/ACCESS.2022.3156278.

Edge AI Technology Report 2023, Published by Wevolver and TinyML.

A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi and J. Kepner, "AI and ML Accelerator Survey and Trends," *2022 IEEE High Performance Extreme Computing Conference (HPEC)*, Waltham, MA, USA, 2022, pp. 1-10, doi: 10.1109/HPEC55821.2022.9926331.

Maxim Micro SDK (MaximSDK) Installation and Maintenance User Guide

Cortex™-M4 Devices Generic User Guide.

A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi and J. Kepner, "Survey of Machine Learning Accelerators," *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, Waltham, MA, USA, 2020,

Cardarilli, G.C., Di Nunzio, L., Fazzolari, R. *et al.* A pseudo-softmax function for hardware-based high speed image classification. *Sci Rep* **11**, 15307 (2021).

Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S. Emer. (2022) Efficient Processing of Deep Neural Networks, A Publication in the Morgan & Claypool Publishers series SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE.

Colby Banbury, Chuteng Zhou, Igor Fedorov, Ramon Matas Navarro, Urmish Thakker, Dibakar Gope, Vijay Janapa Reddi, Matthew Mattina, Paul N. Whatmough,(2021) MICRONETS: NEURAL NETWORK ARCHITECTURES FOR DEPLOYING TINYML APPLICATIONS ON COMMODITY MICROCONTROLLERS, Proceedings of Machine Learning and Systems,

L. Lai, N. Suda, and V. Chandra, "CMSIS-NN: Efficient neural network kernels for arm cortex-M CPUs," 2018. [Online]. Available: arXiv:1801.06601,

Arthur Moss, Hyunjong Lee, Lei Xun, Chulhong Min, Fahim Kawsar, Alessandro Montanari. 2022. Ultra-low Power DNN Accelerators for IoT: Resource Characterization of the MAX78000. In Workshop on Challenges in Artificial Intelligence and Machine Learning for Internet of Things (AIChallengeIoT).

Afshin Niktash, Developing Power-optimized Applications on the MAX78000, Application Note

L. Gwennap, "Kendryte Embeds AI for Surveillance," mar 2019. [Online]. Available: https://www.linleygroup.com/newsletters/newsletter detail.php?num=5992

"GAP Application Processors," 2020. [Online]. Available: https://greenwaves-technologies.com/gap8 gap9/

"SolidRun, Gyrfalcon Develop Arm-based Edge Optimized AI Inference Server," 2020. [Online].

J. McGregor (2020), "Perceive Exits Stealth With Super-Efficient Machine Learning Chip For Smarter Devices,"

S. Han, Q. Gao, C. Wang and J. Zou(2021), "Animal face classification based on deep learning," 2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), Zhuhai, China, 2021,

Faizal, Sahil, Sundaresan, Sanjay (2022), "Wild Animal Classifier Using CNN" International Journal of Advanced Research in Science, Communication and Technology – Publication date: 2022/09/18, doi: 10.48175/IJARSCT-7097.

Mitchell Clay, Christos Grecos, Mukul Shirvaikar, Blake Richey (2022), "Benchmarking the MAX78000 artificial intelligence microcontroller for deep learning applications," Proc. SPIE 12102, Real-Time Image Processing and Deep Learning 2022.

Binta Islam, S.; Valles, D.; Hibbitts, T.J.; Ryberg, W.A.;Walkup, D.K.; Forstner, M.R.J. (2023) Animal Species Recognition with Deep Convolutional Neural

Networks from Ecological Camera Trap Images. Animals 2023, 13, 1526.
Ibrahim Mai, Gebali, Fayez, Li, Kin Fun, Sielecki, Leonard, - Animal Species Recognition Using Deep Learning, 523-532, Publication date: 2020/03/28 doi: 10.1007/978-3-030-44041-1_47
Zualkernan, I. Dhou, S.; Judas, J. Sajun, A.R. Gomez, B.R.; Hussain, L.A.(2022) An IoT System Using Deep Learning to Classify CameraTrap Images on the Edge. Computers.
Miao, Z., Gaynor, K.M., Wang, J. et al.(2019)Insights and approaches using deep learning to classify wildlife. Sci Rep 9, 8137 (2019). https://doi.org/10.1038/s41598-019-44565-w
Lin, W.; Adetomi, A., Arslan, T. (2021) Low-Power Ultra-Small Edge AI Accelerators for Image Recognition with Convolution Neural Networks: Analysis and Future Directions. Electronics 2021, 10, 2048.
Karen Simonyan and Andrew Zisserman (2015) Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations.