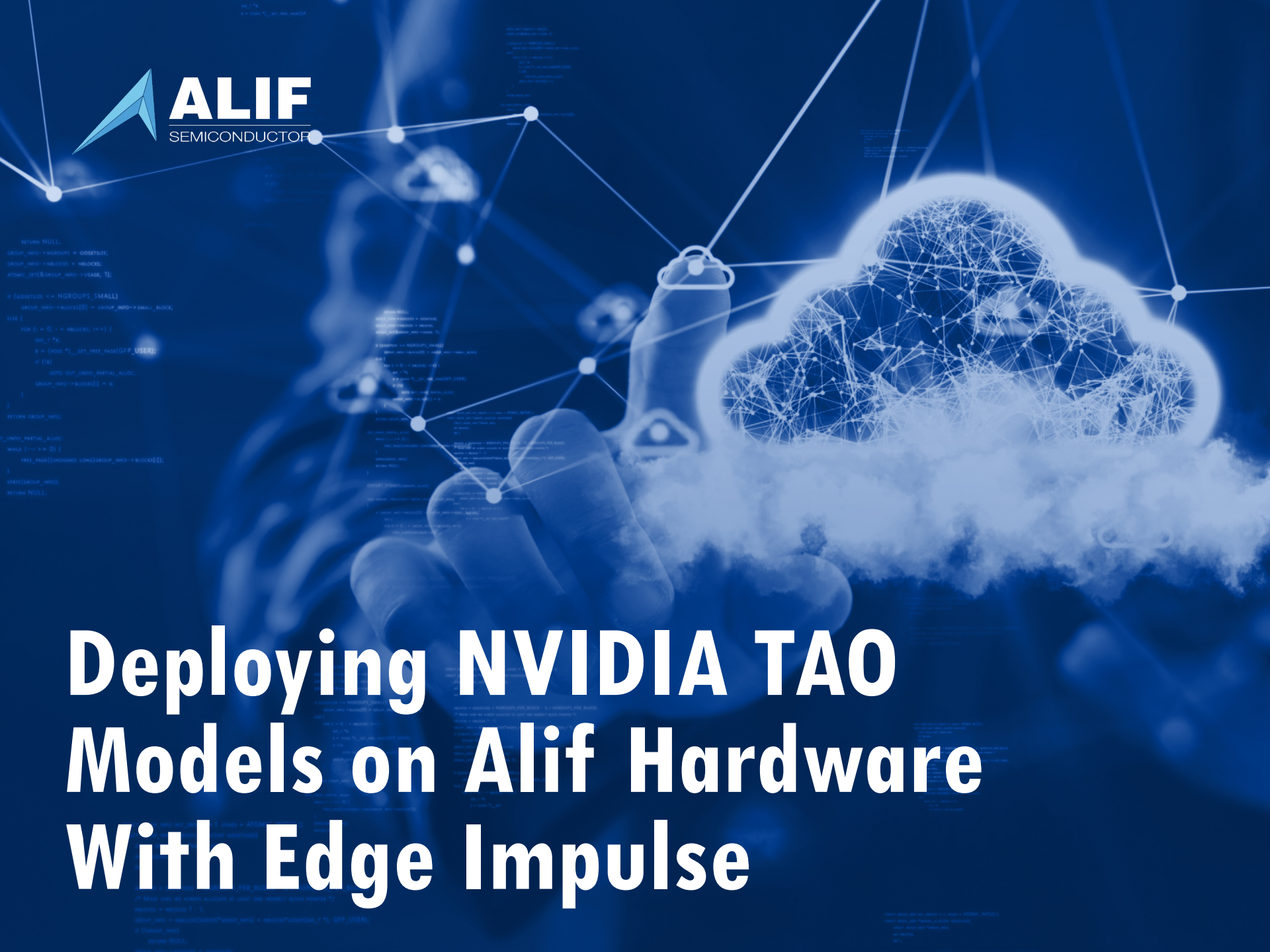


Deploying NVIDIA TAO Models on Alif Hardware With Edge Impulse

Photo credit: Adobe Stocks

Introduction

The NVIDIA TAO toolkit enables embedded engineers to use established and trained Machine Learning (ML) models and modify them slightly to better fit a particular application. This ability allows users to leverage a large library of model architectures for image classification, object detection, and segmentation as well as various task-based models without creating complex models from scratch. While the toolkit concept is appealing and has the potential to massively shorten the design cycle around prototypes, in reality, it has been difficult to deploy these modified models in non-NVIDIA hardware with ease. In theory, other 32-bit MCUs should be able to support these models; however, MCUs without a dedicated NPU, designers are facing an array of challenges.

Edge Impulse has created an IDE for building TAO models and mapping them to supported hardware including a number of GPUs, NPUs, and MCUs. Alif Semiconductor's Ensemble® family of microcontrollers, offers a dedicated NPU (Arm® Ethos™-U55) to run complex AI/ML workloads and has partnered with Edge Impulse to allow engineers to map and accelerate their modified ML algorithm on Alif's hardware.

Moving Compute From the Cloud to the Edge

Traditionally deep learning models are deployed in the cloud, where massive banks of servers are able to adequately train models and process incoming data. Many IoT applications have leveraged AI and the cloud to wirelessly shuttle data from end-nodes so the information could be processed entirely in the cloud. However, this general "sensor-node-to-gateway-to-cloud" architecture has limitations. Depending upon the application, the power-constrained end node might have to continuously transmit wirelessly; this process is power-hungry and will inevitably drain the often battery-powered device. Moreover, the round-trip time taken from the edge (end-device) to the cloud and back is quite high, causing large latencies that may be unacceptable—especially in more safety-critical applications.

Edge computing attempts to shift some of that compute burden from cloud back to the edge device to optimize both the power and timing constraints. This shift could, for example, be used for more autonomous robotics that require more immediate processing/feedback such as drones, autonomous vehicles, automated guided vehicles (AGVs), and more. Other applications could include earbuds that locally process complex audio DSP such as wake word detection, keyword detection, noise cancellation, etc.

Using inference at the edge device to more rapidly run data points into the ML model and calculate the necessary outputs can prove challenging. The process relies on an established and trained ML model—the larger the data set used to train the ML model, the more accurately it accomplishes its specific tasks. However, building a large dataset and labeling images in, for example, object detection models, is a time-consuming and cumbersome task. One solution to optimize this process relies on platforms such as NVIDIA TAO.

Photo credit: Adobe Stocks

What is NVIDIA TAO?

NVIDIA TAO—Train Adapt Optimize—leverages transfer learning by using existing pretrained, weighted AI models with custom data to adapt to a customer's specific synthetic data. One example is using an established object-detection model that typically performs on images in well-lit environments, and optimizing it to function well with custom infrared images **(Figure 1)**.

The use of transfer learning weights side steps the process of building a deep learning model from scratch, a process that typically involves large-scale data collection with millions of data points, labeling, and training to effectively fine-tune a model. The toolkit is built on TensorFlow and PyTorch to train, fine-tune, prune, quantize, and export more purpose-built models. The NVIDIA TAO provides a simple command line interface to train a deep-learning model for computer vision (CV) tasks including classification, object detection, instance segmentation, and semantic segmentation.

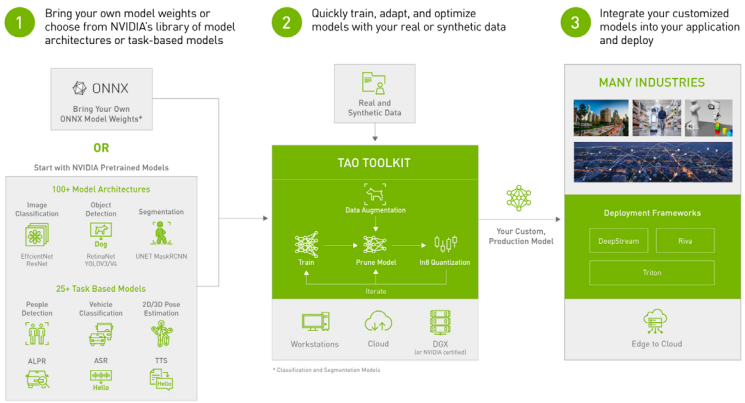


Figure 1: The TAO workflow diagram involves data, training, and deployment.

The Problem: Deploying ML on 32-bit MCUs

The application space for using ML at the embedded edge is growing, and new platforms are being deployed to support this with various hardware including ASICs, MPUs, GPUs, FPGAs, CPUs, MCUs, DSP chips, and many multi-core architectures as well. Embedded developers generally prefer running their programs on MCUs for ease of programming, small form factor, and reduced power consumption. However, there are factors that limit the average 32-bit MCU's ability to seamlessly run ML inference. MCUs and the cores within them will execute tasks serially, making the processor face several bottlenecks in performance (e.g., memory wall, Von Neumann). Moreover, the increase in time it takes to process and receive an "answer" causes the device to remain "on" for longer periods of time and unable to revert back to a low-power sleep mode. The increased processing time, in turn, yields a sharp increase in power consumption—an unacceptable outcome in many battery-powered applications.

It is a challenging task to migrate models down to operate efficiently on the 32-bit MCU architecture. And, while the low-code, open-source workflow of NVIDIA TAO is appealing and has the potential to help with this issue, these models were historically difficult to run on anything other than NVIDIA hardware— and difficult to run efficiently on 32-bit MCUs.

Using Edge Impulse to Deploy TAO Models

Edge Impulse gives a complete integrated development environment (IDE) for building TAO models including data collection, training and validating models, and deployment to a wide range of devices from MCUs to higher-end GPUs. The entire platform offers firmware ready to be flashed onto the hardware of choice as well as established binaries for fully-supported development boards. Edge Impulse offers the tools to create digital twins, synthetic datasets, and virtual model testing environments.

Alif Semiconductor has partnered with Edge Impulse so that users, from engineers with limited ML knowledge and programming ability to the more seasoned developers, could use the Edge Impulse Studio or Python SDK to deploy their fine-tuned ML model to the Alif Ensemble MCU. This Ensemble series offers dedicated NPUs (Ethos-U55) to run complex AI/ML workloads.

The Benefits of the Alif's Ensemble Family of MCUs

The Ensemble family of MCUs starts with the E1, a single-core Arm® Cortex®-M55 with an optional dedicated Ethos-U55 microNPU for accelerating ML workloads. As shown in Figure 2, the number of cores can be scaled up while also introducing cores that run at higher clock frequencies to process data more quickly. The E5 and E7 are known as fusion processors that combine MCUs, NPUs, and MPU(Cortex-A32) in a single package. The more cores, the faster the ability of the MCU to process the ML tasks. Alif has introduced NPUs that can perform parallel computations instead of serializing math through layers, including the most basic Ensemble MCUs.

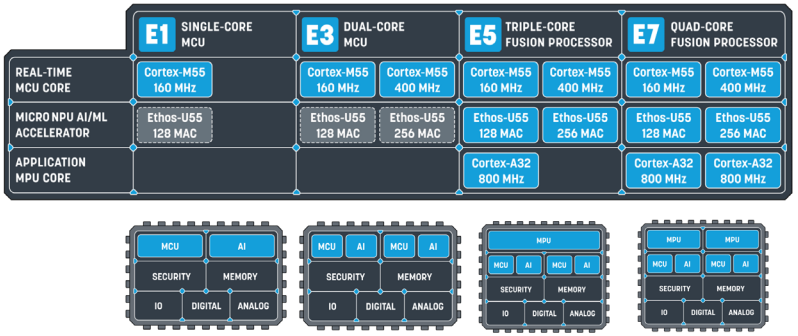


Figure 2: The Alif Semiconductor Ensemble family of MCUs.

Since Alif Semiconductor is using modern technology, the Ensemble family can afford to integrate much more memory; the traditional MCU typically carries ~256 kB of RAM whereas the Ensemble carries up to 14 MB of RAM on die. This advantage loosens the constraints that designers typically face when mapping their ML algorithms to hardware. Other Alif families such as the Balleto™ family of MCUs will also incorporate the Ethos U55 NPU while also integrating BLE 5.3 and IEEE 802.15.4 radio models, removing the typical two-chip solution for most solutions that incorporate wireless transmission. All of these SoCs will also host a wide variety of peripherals to better suit the embedded edge application.

The combined benefits push ML tasks to run faster on an Alif device. Speed is an important aspect of the process, since inference workloads on MCUs will generally force it to run at 100% utilization, driving up power consumption. Shortening the amount of time it takes to get an "answer" allows the device to quickly switch back to the desired low-power mode to extend battery life. Battery life is a uniquely challenging aspect of portable edge devices and wearables with small form factors such as fitness trackers, sleep monitoring devices, and wireless earbuds.

The challenge in these applications is not only finding the right hardware to run inference efficiently, but procuring an integrated solution that suits the end-applications form factor.

Using Edge Impulse Studio to Run a License Plate Detection Algorithm on the Alif Ensemble MCU

The partnership with Alif Semiconductor and Edge Impulse supports, for instance, creating a license plate detection and decoding algorithm on the Alif Ensemble MCU. The NVIDIA TAO toolkit has a number of models that can be fine-tuned for this specific purpose. The TrafficCamNet and DashCamNet models can be used to detect vehicles, while the multi-purpose detection (MPD) mode can be used to detect license plates, and the license plate recognition (LPR) model, to translate images to text. The challenge is to make sure the algorithms work in the images' specific environment. There are a number of factors that could cause the ML model to fail:

- Car velocity: the vehicle could be stationary or moving
- Camera positioning: the camera could be positioned on a pole at different angles
- Time of day: Depending upon where the sun is, the camera may receive glare that prevents it from visualizing the license plates passing through its field of view (FoV)

The model might need to be adapted for these factors to obtain the level of accuracy required. In order to do this users are able to create a synthesized dataset. The general workflow can be seen in **Figure 3**.

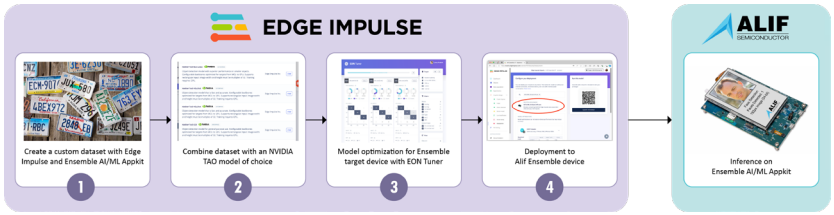


Figure 3: The workflow to train a license plate detection model using the NVIDIA TAO YOLOV3 pre-trained model integrated into Edge Impulse.

Create a Custom Dataset

After the board is connected and users login to the Edge Impulse Dashboard, users can begin creating a custom dataset by clicking "Dataset." "Data collection" can begin by capturing a series of images of license plates via a smartphone, webcam, or a connected Ensemble AI/ML Appkit and manually labeling some, but not all, of these images. Users that already have images can upload them by clicking on "Data acquisition" and adding data. The bigger the dataset and the more accurately labeled it is, the more accurate the model will be with the given camera positions, lighting, and vehicle velocities.

The model is then put together under "Impulse design," where "Image data" will include the new images for pre-processing. At this stage, users can choose two "blocks:" an image processing block and an object detection learning block. Other features can be entered in this step, such as a post estimation block to train a classifier. In the learning block, the NVIDIA TAO model has the right architecture to do that (**Figure 4**).

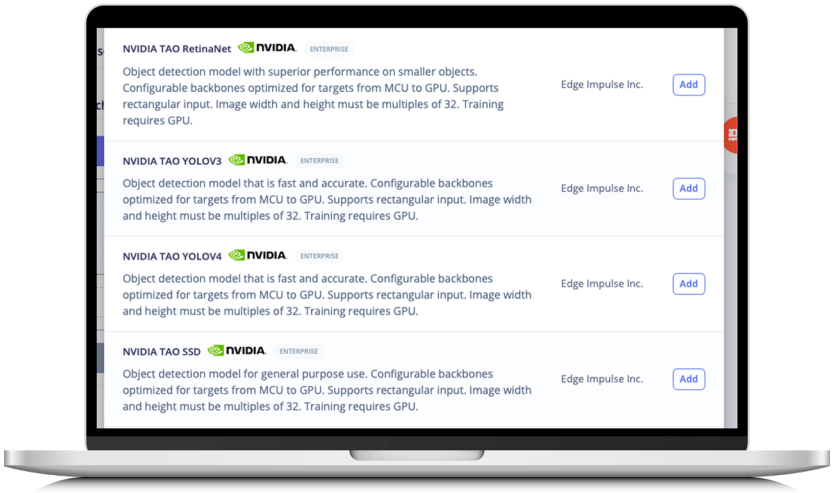


Figure 4: Edge Impulse window for selecting the NVIDIA TAO model.

Navigating to the "Object detection" button will lead to the various neural network settings that can be modified. These include parameters such as "Number of training cycles," "Learning rate," and α. Finally, when it comes time to train the model, Edge Impulse will spin up a GPU in its cluster and load the NVIDIA TAO containers to combine them with the custom dataset that was originally created. This process is done on the cloud in order to train the algorithm more efficiently.

At this point, there are two ways to view the data: in a 2D map under "Feature explorer" or as "On-device performance." The "Feature explorer" uses the embedding of the network to visualize clusters of data that show obvious similarities, e.g., license plate by state, license plate by number, similar vehicle, etc. Furthermore users can view the estimated inference time as well as RAM and flash usage under "On-device performance" (**Figure 5**).

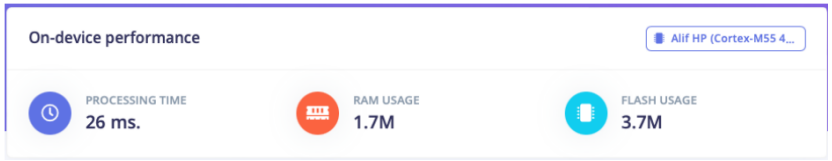


Figure 5: The Edge Impulse on-device performance shown when running inference on the connected Alif AI/ML kit.

Finding the Best ML Model for the MCU

Further optimizations can occur with an AutoML tool called the "EON Tuner." This tool enables users to find the most optimal embedded ML model for the application within the constraints of the Alif target device. **Figure 6** shows the EON Tuner tool being used to find the most ideal model to run a keyword spotting algorithm on a Cortex-M4 running at 80 MHz including its respective memory constraints.

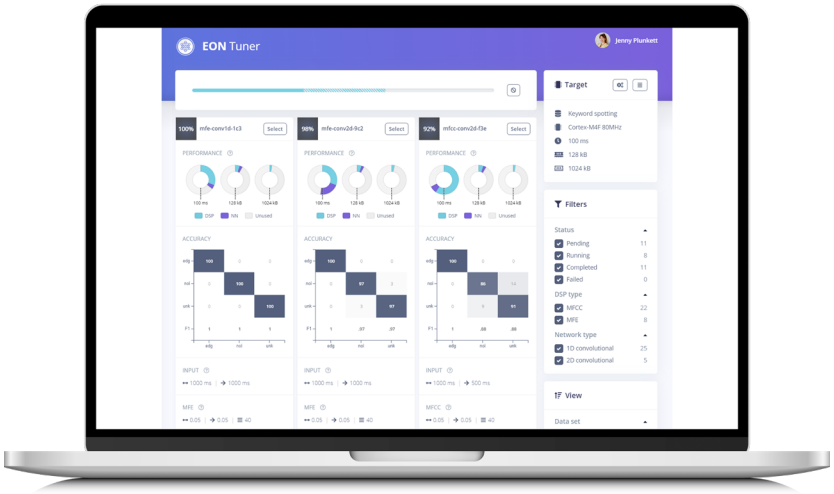


Figure 6: EON Tuner tool showing the best NVIDIA TAO models available to run most efficiently on the target device.

Deployment to Alif Ensemble Device

The user can tune the model for the underlying MCU architecture: the Alif Ensemble MCU with an embedded NPU. This key step allows users to deploy directly to the Alif MCU target device with zero external dependencies (**Figure 7**). A number of deployment options are available for Alif hardware on Edge Impulse. Once deployed, Edge Impulse will provide a .zip file that contains the flashing utility and firmware for the Ensemble AI/ML Appkit.

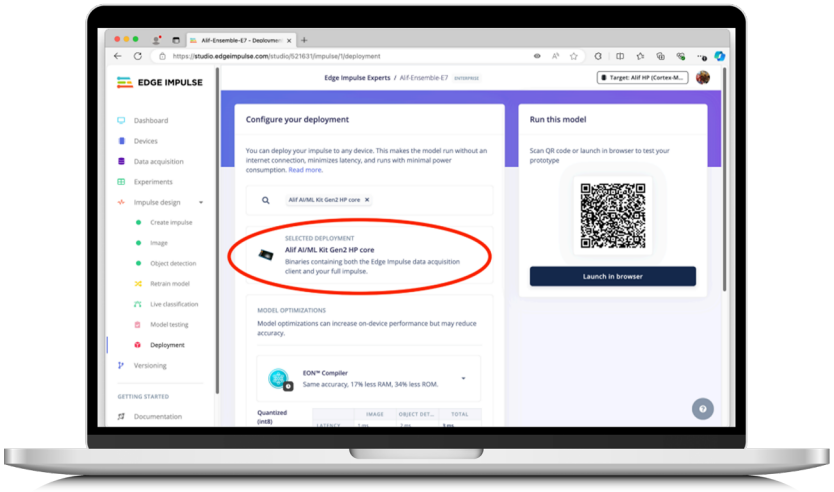
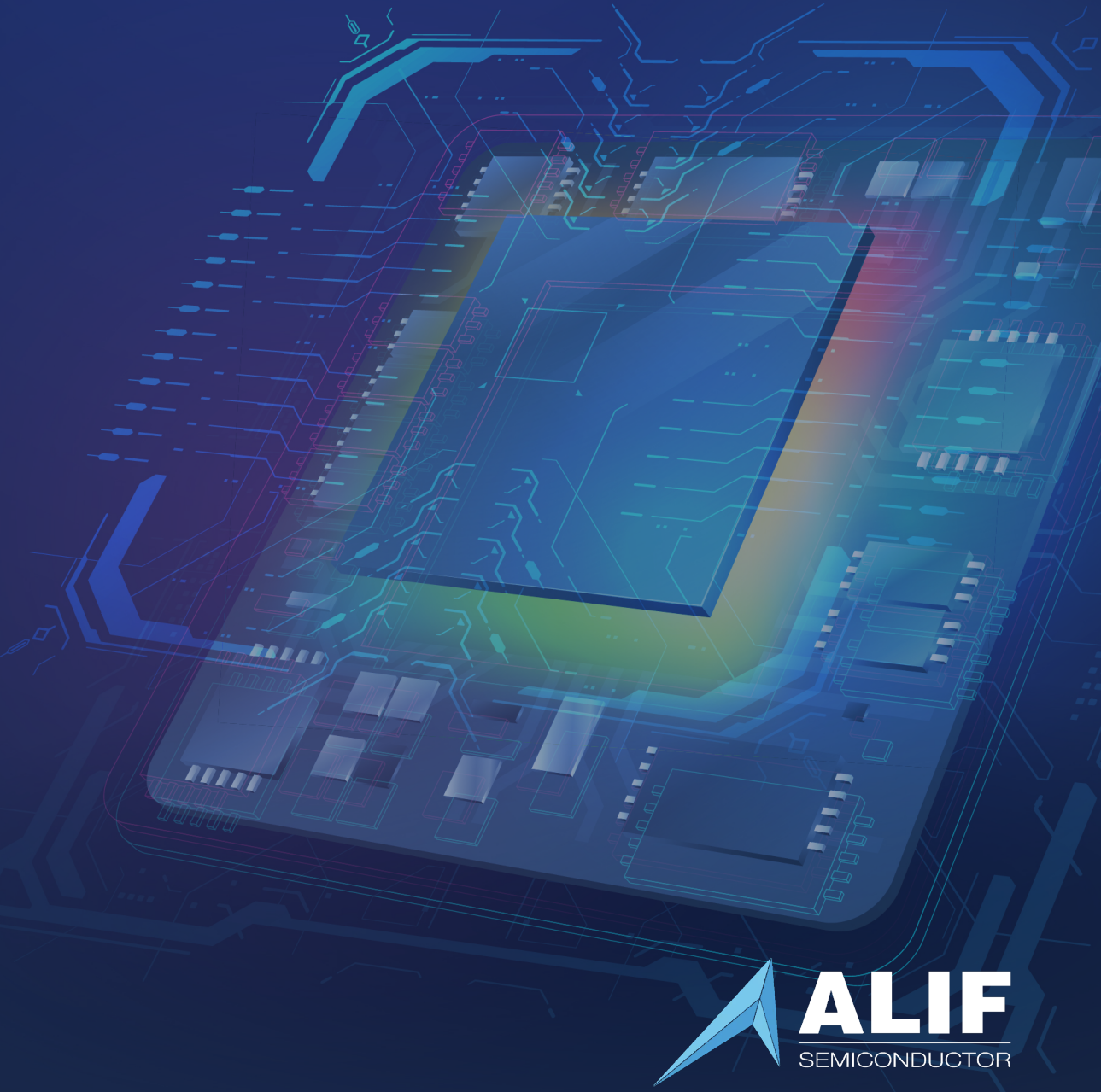


Figure 7: Deployment to the Alif AI/ML Kit Gen2 HP core via the Edge Impulse Studio.

Photo credit: Adobe Stocks

Conclusion

More and more applications are cropping up that require computing at the edge. In order to adequately process these complex ML algorithms, it is critical to have the hardware that can accelerate these workloads. However, this acceleration brings a challenge: many solutions that allow for parallel compute capability are often too large and are too catered for a particular application (ASIC). MCUs offer many advantages for embedded engineers, but it is difficult to perform inference on these devices without running them too hard. Alif offers a series of MCUs with multiple cores, a wide range of peripherals, and embedded NPUs to run ML workloads as efficiently as possible in an integrated MCU platform. The partnership with Edge Impulse allows users to leverage pretrained models and adapt them to their specific edge task with an intuitive GUI.



ALIF
SEMICONDUCTOR