element14 ESSENTIALS

Artificial Intelligence II: AI at the Edge

Sponsored by Intel, Molex, NXP Semiconductors, ST Microelectronics, & Xilinx

1. Introduction

Artificial intelligence (AI) has, for many years, operated within data centers in the Cloud, which had the necessary power to perform compute-intensive tasks that AI algorithms require to process vast amounts of data. But the latency, privacy, security, and bandwidth challenges associated with cloud computing, especially for time-sensitive decision making, has spurred the evolution and development of AI at the Edge of the network. Edge devices solve these challenges by processing data locally and close to the source of the data. Edge devices and gateway-to-edge devices are now more powerful, and this enables local collection, storage, and analysis of data without waiting for the results to be derived from the Cloud and then passed back to the device. Diverse applications employ functional Artificial Intelligence capabilities, including video, audio, and image analysis. This learning module explores AI at the Edge with respect to basic concepts, hardware, and design tools of interest to a designer.

2. Objectives

Upon completion of this module, you will be able to:

- Define AI at the Edge as it relates to Edge Computing and AI
- Describe the AI platforms found in Edge AI
- Explain how latency, bandwidth, availability, security, and privacy are critical factors in edge computing
- Discuss the development tools used for AI at the Edge solutions

3. Scope

Once the basis of storylines for sci-fi movies, Artificial Intelligence (AI) now has practical applications that are changing the way businesses operate and the way consumers use technology in their everyday lives. Developers are exploring ways to combine AI with the Internet of Things (IoT) to help companies benefit

from the data generated by connected devices. In 2018, Forrester Research found that 48% of companies in North America had already invested in AI/machine learning as part of their digital strategies.

When developing solutions, it is essential to consider what infrastructure best supports the ability of AI to drive real-time decision making. While cloud solutions have historically been widespread, latency matters, and waiting on data centers miles away to power instant decision-making is not feasible for many applications. Edge computing is the answer in many cases. Emerging advancements in Edge hardware and modules have helped to push progress in AI at the Edge of the network. By combining AI and edge computing, IoT solutions are more potent as the latency issues associated with cloud computing are eliminated.

While AI at the Edge is a vast topic to discuss, in this learning module, we will focus on its essentials, including definitions, basic concepts, and components, as well as AI development tools in edge computing.

4. Basic Concepts

In Edge AI, the AI algorithms are processed locally on a hardware device. The algorithms use data (signals or sensor data) created on that device. An Edge AI-equipped device need not be connected to a network in order to work correctly, as it can process data and make independent decisions sans any connection. The device must be fitted with a microprocessor and sensors for the collection of data for AI implementation.

A handheld IoT-enabled power tool is a good example. It is by definition on the Edge of the network. Data acquired by the microprocessor operating on the tool is processed by the Edge AI software application in real-time. The Edge AI application generates results and stores the same locally on the device. The power tool, after working for hours, connects to the internet and sends the data to the Cloud for storage and subsequent processing. Extended battery life is a must in such cases. The battery would be drained too quickly if the power tool were to continuously stream data to the Cloud.

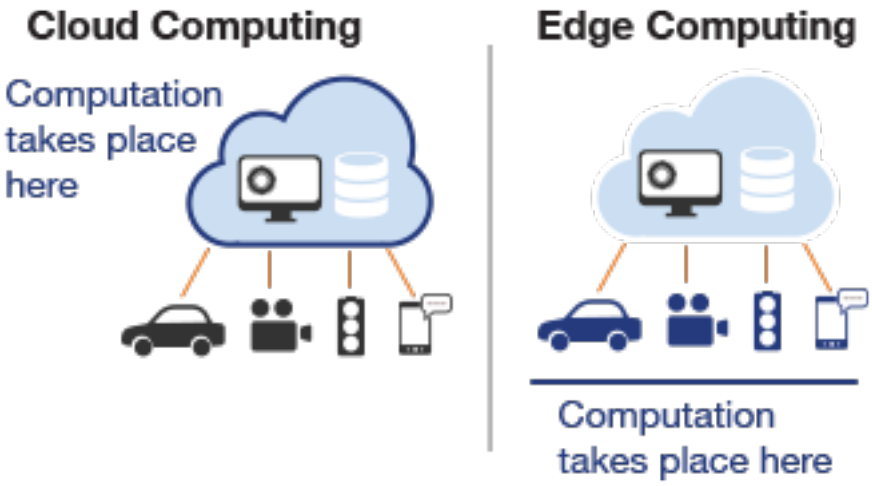

Figure 1: Cloud computing vs Edge computing

Edge computing shifts computing to near the data source, instead of relying on one of any dozen Cloud data centers. Cloud computing, built on a centralized processing model, locates the remote run of workloads over the internet inside a commercial provider's data center. It tasks the network with uploading all data to the cloud data center and uses its supercomputing power to solve computing and storage complications. Conversely, edge computing is based on a de-centralized process model. It is crucial in solving possibly disastrous situations such as:

- **Collisions:** Autonomous vehicles cannot tolerate even a minimal delay between sensing any possible collision and making the needed adjustment, like steering away from trouble or applying the brakes to decelerate or stop.

- **Fire:** In industrial safety systems like smoke detectors and fire alarms, a few seconds squandered in data transmission may turn out to be catastrophic.

- **Environmental Hazards:** Near-instant data capture and subsequent analysis at oil well sites, for example, may accurately predict disasters and commence tragedy-preventive measures.

In all the preceding instances, the wait time for data to travel to the Cloud and back may be fatal. Low-latency edge computing is critical.

Edge computing is superior to cloud computing for innumerable reasons, including:

- **Latency:** Novel applications in IoT come with high real-time needs. Applications, in the traditional cloud computing model, dispatch data to the data center, and the subsequent response multiplies system latency. The edge AI

system is primarily an independent system with no need for connectivity to the Cloud. This reduces system latency and improves service response capability.

- **Bandwidth:** An enormous amount of data, generated by edge devices, can bog Cloud services from sensors capturing and transmitting data in real-time. Data processing taking place at the Edge of the network, with essential output parameters being uploaded to the Cloud, can substantially reduce pressure on data center power consumption and network bandwidth.

- **Availability:** With internet services increasingly deployed on the Cloud, its availability is now a given in daily life. A few illustrative examples are voice-based applications like Apple's Siri and Google's voice assistant. If connectivity is lost, the users may experience difficulty in using them, and hence cloud service providers thus struggle to maintain 24x7 availability.

- **Security and Privacy:** Data generated from interconnected devices relate closely to users' lives. The private information of users can be compromised when indoor cameras transmit house video data to the Cloud, for example. Edge computing does not upload users' private data, but instead stores it on edge devices, thus reducing network data leakage risk. Privacy and security remain protected.

AI and Edge Computing will converge in time. This is natural and inevitable, as an interactive relationship exists between the two. AI offers methods and technologies, and Edge Computing brings scalability and potential. Conversely, Edge Computing provides platforms and scenarios to AI, and the latter can diversify its applicability with Edge Computing.

Edge Computing is a typical example of distributed computing, where software-defined networks decentralize data and offer robust, elastic services. It encounters resource allocation complications in different layers, like CPU cycle frequency, radio-frequency, and access jurisdiction. Substantial demands are consequently imposed on several powerful optimization tools to increase system efficiency. AI technologies can manage this task.

5. Analysis

AI chips with computational acceleration like Field Programmable Gate Arrays (FPGAs), Neural Processing Units (NPUs), Graphics Processing Units (GPUs), and Tensor Processing Units (TPUs) can now be integrated with intelligent mobile devices. An increasing number of corporations participate in chip architectures design to support typical edge computation and facilitate Deep Neural Network (DNN) acceleration on resource-constrained IoT devices. The Edge hardware upgrade also injects vigor and vitality into AI. Semiconductor manufacturers such as Xilinx, NXP, STMicroelectronics, and Intel have released

powerful edge processors with low energy needs, including software kits for the implementation of AI.

- ***5.1 Implementation System***

The implementation system undergoes three distinct phases:

1. Collation of Data and Model Selection: A user, in this discovery phase, uploads and labels training data, and tests different machine learning (ML) models. Several models can be obtained from multiple libraries. The most useful models for that specific domain are discovered.

2. Training: The user collects data and creates a dataset from which the models are trained. The algorithm is trained, and during training, modifies the training parameters. The data is also modified and yields an output that has to be evaluated and checked. Training should be continued otherwise. The model is automatically retrained multiple times. The training steps have requirements in terms of software and hardware, with a faster GPU being essential in the training phase.

3. Deployment: Models created during the training phase can be deployed. These models have a smaller binary size, better performance, and fewer dependencies. A faster GPU is not needed, and software that was necessary for training steps can easily be deployed on the edge devices.

- 5.2 Reducing Latency at the Edge

AI at the Edge has various use cases in supporting industrial automation, automotive, medical, and aerospace and defense industries with established safety and reliability standards. This technology can be used in motion control, M2M communication, predictive maintenance, smart energy and smart grid, Big Data analytics, and intelligent connected medical systems. Automotive edge applications include forward camera and surround view systems and imaging RADAR and LiDAR sensors.

AI object detection algorithms are widely used in the ADAS, medical, and smart city domains. Single-shot object detectors are the favored type of algorithm. Single Shot Detector (SSD) is an easily trainable neural network model that has superior accuracy even with a smaller input image size. You Only Look Once (YOLO) is a state-of-the-art and real-time object detection system. All these neural networks are deep neural networks and come under deep learning.

The hardware used for edge AI systems can be challenging, as it must satisfy needs such as power efficiency, discreet form factors, and peripherals that can

be implemented on hardware to software on a single device. The optimized hardware acceleration of AI inference and other performance-critical functions reduces latency at the Edge. The Edge solution stack is geared towards functional safety, real-time vision and control, industrial networking, machine learning (ML), mixed-criticality software domains, and robust cyber security anchored in a hardware-root-of-trust without compromising on performance/watt, reliability over harsh conditions, and longevity of supply.

- ***5.3 Hardware for Edge AI***

For an AI model to run efficiently, the hardware used must have sufficient processing power. Xilinx solutions offer an application processor controller (quad Arm Cortex-A53) partnered with a Deep Learning Processor Unit (DPU), a co-processor IP. This solution has the flexibility to receive and format a broad swathe of sensor data. The ecosystem offers a different approach to solve edge applications involving use cases such as predictive maintenance, digital twin model-based control, and anomaly detection. ML frameworks support Caffe, Darknet, and TensorFlow. Pre-trained, optimized models are available at high-level abstraction for targeted embedded applications.

Re-programmability is indispensable to all Xilinx technology, and combined with adaptive silicon, allows domain-specific architectures to be updated, optimizing to the newest models without the need for new silicon. The Ultra96™ makes a viable platform for building edge use-case ML applications. This heterogeneous device enables us to correctly engineer a solution that achieves the balance between power and performance, which are critical parameters for edge-based applications.

NXP also has Application processors which target high-end AI at the edge applications, such as the IMX 8, 7, and 6 series, and for low-end applications the i.MX RT Crossover MCUs as well as LPC5500 Series microcontrollers.

ST Microelectronics (STM) focuses on low-end Edge AI applications and has introduced various microcontrollers for these, such as the STM32 F4, L4 or L7 families based on the Cortex m4 and m7 microcontrollers. The STM32 Arm® Cortex®-A7® microprocessor series and STM32 Arm® Cortex® M4/M33/M7-based microcontrollers with floating point capabilities can work on sensor data at the Edge.

STM has also introduced sensors such as the LSM6DSOX, a 3D digital accelerometer and 3D digital gyroscope with an ML core that does not need an additional controller to process machine learning functions. It has better accuracy capabilities for consumer electronics, wearable technology, battery-operated IoT, and gaming.

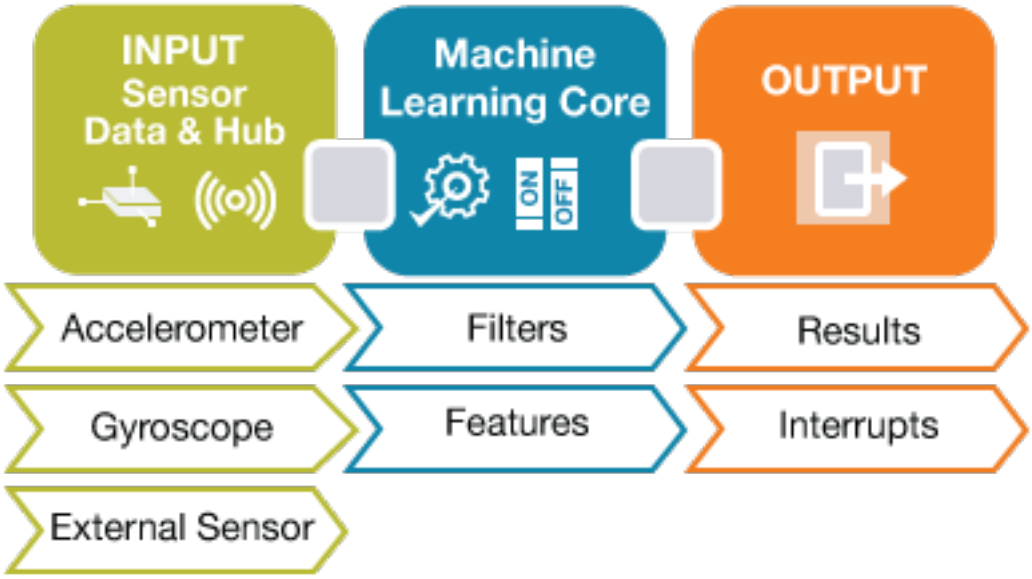

Figure 2: Machine Learning Core the in ST Microelectronics LSM6DSO

Powerful computing is required to serve more data and new capabilities at the edge. Solutions derived from Intel® platforms deliver this performance, and specialized technologies are used to consolidate applications and data onto a common platform while appending AI and many other functionalities. A simpler IT environment helps businesses to reduce their costs. Intel offers the Atom IoTG, Core IoTG, Xeon IoTG, and other similar processors for system development.

- ***5.4 Development and Deployment Tools***

In this section we will cover some of the development and deployment tools used for AI at the Edge.

The Xilinx Edge AI platform: Delivers comprehensive tools and models which use unique deep compression and hardware-accelerated Deep Learning technology. The AI solution consists of a Data Center AI Platform and an Edge AI platform. Edge AI is obtainable on a Zynq SoC and MPSoC Edge card, supported by the Deep Neural Network Development Kit (DNNDK) toolchain. The DNNDK optimizes the trained model, and provides a solution that combines software programmability, real-time processing, hardware optimization, and any-to-any connectivity with the security and safety needed for Industrial IoT systems. Xilinx SDAccel™, SDSoC™, and Vivado® High-Level Synthesis enable customers to quickly develop smarter connected and differentiated applications. Python-powered control, edge analytics, and ML-enabled by PYNQ- a software-hardware framework for Zynq SoCs - controls the programmable hardware to pre-process sensor data and other types of data to

make software analysis and manipulation ultra-efficient in an embedded processor.

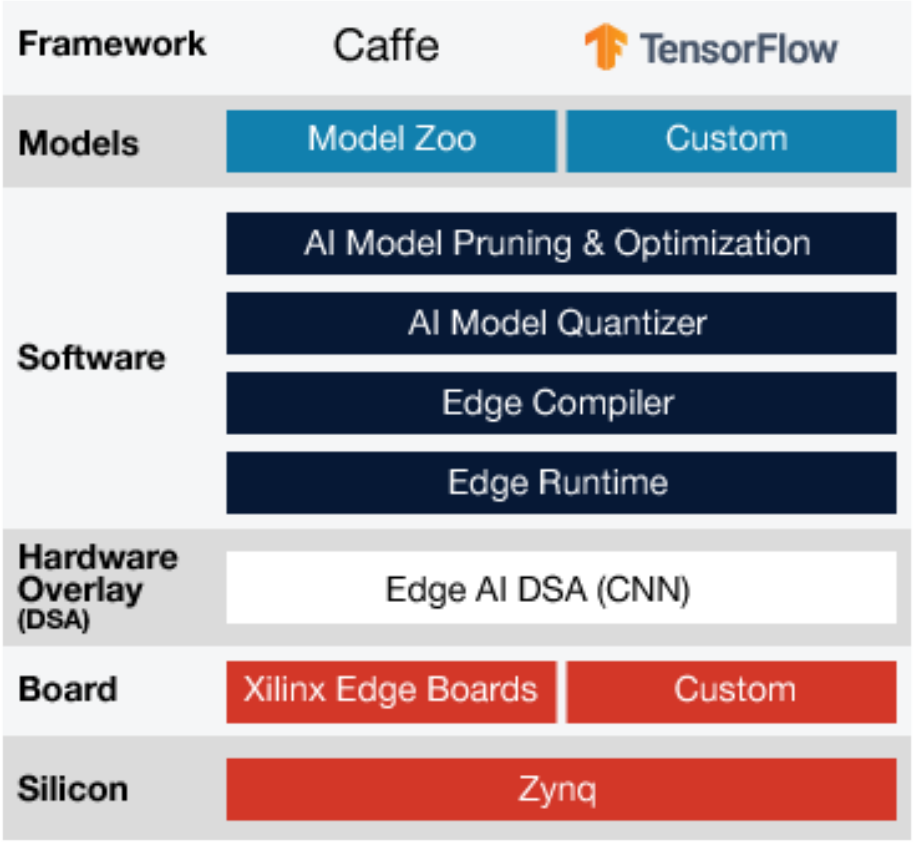

Figure 3: Xilinx AI Edge Platform

eIQ™ ML Software Development Environment: NXP Semiconductors developed eIQ™ ML Software Development Environment enables ML algorithms for use on NXP MCUs, i.MX, and RT crossover MCUs SoCs. eIQ software includes inference engines, optimized libraries, neural network compilers, and optimized libraries. This software is fully integrated into the MCUXpresso SDK and Yocto development environments, allowing secure development of complete system-level applications.

Click to enlarge image

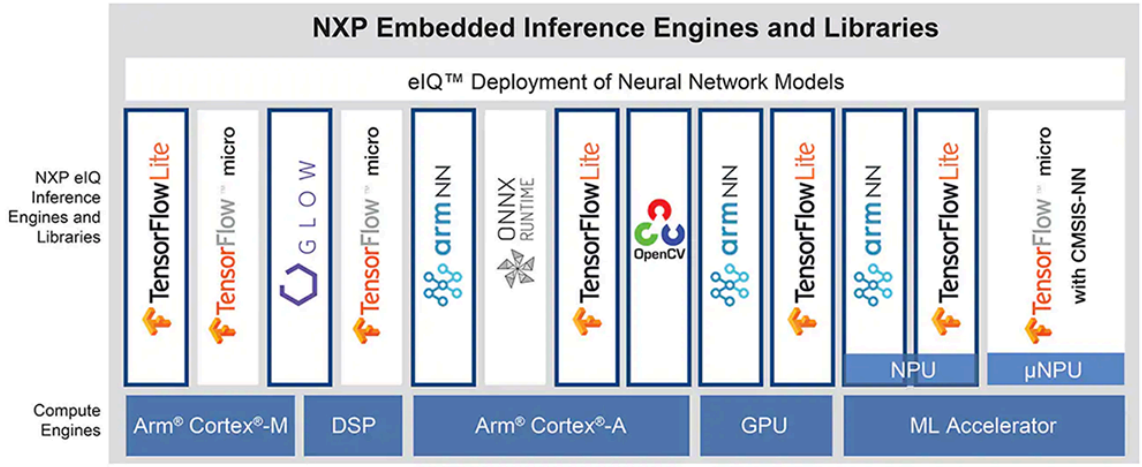

Figure 4: NXP Inference Engines and Libraries

Software support for eIQ ML Software Development Environment include:

- **TensorFlow™ Lite:** TF Lite defines a FlatBuffers based model file format. The FlatBuffers memory footprint is of a smaller order of magnitude, permitting better cache lines to use, leading to quicker execution on the NXP devices.

- **Arm®NN:** This is an inference engine framework that offers a bridge between Arm machine learning processors and neural network (NN) frameworks, including NXP's Layerscape® and i.MX processors. Arm NN recasts existing neural network framework models into inference engines which leverage Arm Neon™ instructions through the Arm Compute Library.

- **CMSIS-NN:** This is a neural network kernel collection used for efficient performance maximization and to minimize the neural network's memory footprint on Arm® Cortex®-M processor cores. The NXP eIQ, for easier deployment, integrates the CMSIS-NN directly into the MCUXpresso, along with all other Arm CMSIS components. CMSIS-NN needs more manual intervention compared to TensorFlow Lite, but yields a smaller memory footprint and quicker performance, by sacrificing a limited set of neural network operators.

- **Glow:** The eIQ machine learning (ML) software development environment reinforces the Glow ML compiler, enabling timely MCU compilation for MCUs. This compiler converts neural networks to object files, and the user transforms this into a binary image for a smaller memory footprint and increased performance as compared to a standard runtime inference engine. Glow finds use as a software back-end for the PyTorch machine learning framework. It supports the ONNX model format. Glow, an abbreviation of Graph Lowering,

derives its name as it drops a neural network to a two-phase strongly-typed intermediate representation.

- **OpenCV™:** OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. It helps in executing image processing tasks, object detection, video encoding/decoding, deep neural networks, and machine learning algorithms processing.

X-cube-AI: The X-Cube AI is the extension pack of the STM32cube MX. The STM32CubeMX is one graphical tool which permits STM32 microcontrollers and microprocessors to be easily configured, and also the corresponding initialization C code for the Arm® STM32 cube generation. The AI (aka X cube AI) is forged around the library generator to efficiently convert (after import) pre-trained artificial neural networks that are developed with Keras, Lasagne, and other deep learning (DL) frameworks. ST's X-cube AI sets the map and runs pre-trained Artificial Neural Networks (ANN) using the broad STM32 microcontroller. The deployment of AI to STM32 microcontrollers enables technology at the Edge nearer to the embedded low power MCU devices and sensors.

OpenVINO™ toolkit: The OpenVINO™ toolkit is a comprehensive toolkit for quickly developing applications and solutions that emulate human vision. The toolkit is based on Convolutional Neural Networks (CNNs), and extends computer vision (CV) workloads over Intel® hardware, maximizing performance. It also supports heterogeneous execution across various Intel devices, such as an Intel® CPU, FPGA, or the Intel® Movidius™ Neural Compute Stick. These libraries are easy to use for computer vision functions, which speeds up the end products to be released in the market.

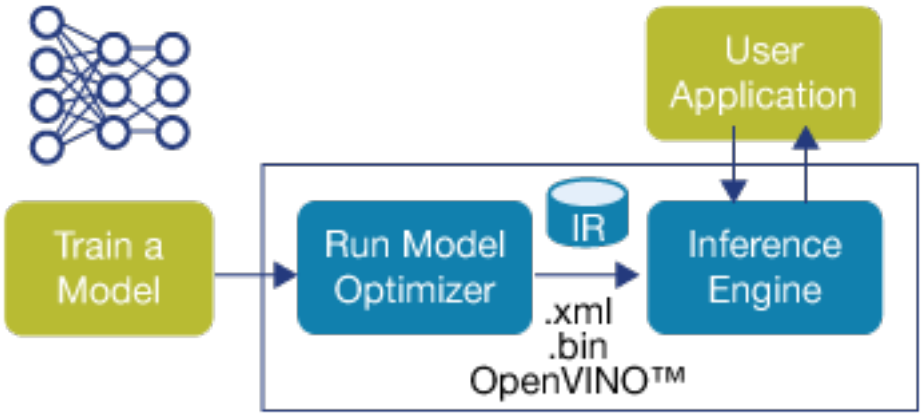

Figure 5: Intel® Deep Learning Deployment Toolkit allows developers to deploy pretrained deep learning models through a high-level C++ or Python inference engine API integrated with application logic. It supports multiple Intel® platforms and is included in the Intel® Distribution of OpenVINO™ toolkit*

The OpenVINO toolkit has three main parts:

- **Inference Engine:** Refers to the software libraries which run inference in opposition to the Intermediate Representation (the optimized model) to generate inference results.

- **Model Optimizer:** This optimizes models for Intel® architecture. The Model Optimizer converts models into an Inference Engine compatible format. Such a format is termed Intermediate Representation (IR).

- **Intermediate Representation (IR):** This is the Model Optimizer output. It is a model reworked to an optimized Intel® architecture format and can be used by the Inference Engine.

- ***5.5 Interconnect Solutions for Edge AI Designs***

Interconnect solutions are vital to AI design, as a system can only be as robust as its weakest signal connection. For example, a chip laboring through an intermittent signal connection will deprive the application of critical data. An inadequate design may trigger interconnection bottlenecks during data presentation. It follows that a better interconnect link correlates to better system reliability. Bandwidth is equally essential. AI applications, notably sensors and cameras, generate huge amounts of data. Interconnect systems such as micro board-to-board connectors, along with industrial products like corsets and M8 or M12 connectors, empower these technologies to produce their best performance without changing the existing AI algorithms. Interconnects also have an important function in the Industrial Internet of Things (IIoT). On the plant floor, interconnect solutions find use in connectivity, robotics, power and motors, communication and control, and wireless applications.

AI is evolving the transportation market. It imports safer and more efficient methods, from driver safety devices, finding parking spots, and traffic control to the truck fleet organization. AI will offer streamlined, more reliable, and cost-saving systems. Automotive and commercial vehicle OEMs must integrate a denser connector and sensor population into advanced driver-assistance system (ADAS) modules to facilitate autonomous driving, while simultaneously condensing PCB footprint. Although Molex does not directly design AI applications, it is a significant contributor as it supplies the signal and board-to-board interconnects that fetch signals and power to the devices and sensors which propel AI systems. Products today need higher bandwidth to send video or data from camera/sensor inputs fitted inside contemporary self-driving vehicles. There is thus a necessity for high-speed signals, and Molex offers products that boost the performance of such devices.

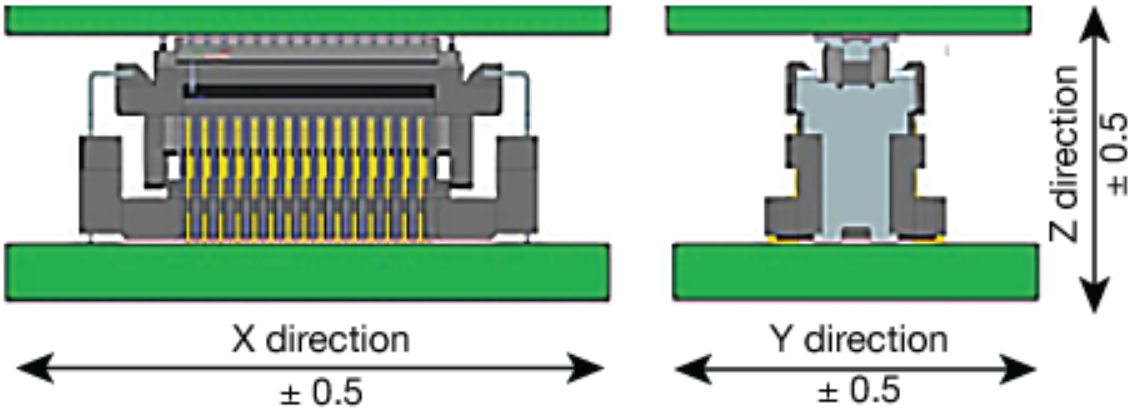

Figure 6: Molex SlimStack Floating Board-to-Board Connectors improve reliability for Edge AI applications as they prevent poor contact reliability and solder cracks by absorbing misalignment stresses.

As transportation market requirements evolve into complicated ecosystems with constricted space needs, designers will require multiple types of cables and connectors that help maximize electrical performance and design flexibility, including in-line, unsealed, and sealed connectors, and in vertical and right-angle PCB configurations. Higher resolution displays on the manufacturing floor need better signal integrity and EMI performance. Robust, high-speed connectors are compulsory to satisfy such high-performance demands. FFC/FPC connectors offer both fast data rates and high reliability. The next installment of features pushed by AI will need higher circuit density for PCB and flex assemblies within the same application profile. This leads to increased modularity, while simultaneously boosting low-profile connector demand.

6. Glossary

- **ADAS:** Advanced driver assistance systems.
- **AI:** Artificial Intelligence.
- **Condition Based Monitoring:** Condition Based Monitoring (CBM) is a type of predictive maintenance that involves using sensors to measure the status of an asset over time while it is in operation.
- **DPU:** Deep Learning Processor Unit.
- **Extension Pack:** An Extension Pack is a set of extensions that can be installed together.
- **FFC:** Flexible flat cable.

- **FPGA:** Field Programmable Gate Array.
- **FPN:** Feature Pyramid Network.
- **GPU:** Graphical Processing Unit.
- **ML:** Machine Learning.
- **NPU:** Neural Processing Unit.
- **Predictive Maintenance:** Maintenance that monitors the performance and condition of equipment during normal operation to reduce the likelihood of failures.
- **SoC:** System on chip.
- **SSD:** Single Shot MultiBox Detector.
- **TPU:** Tensor Processing Unit.

Take the Quiz