Electronic Design

By ART STOUT, Director of Product Management, AI solutions, Teledyne FLIR, https://www.flir.com.

Improving Convolutional Neural Networks at the Edge

Artificial Intelligence and more specifically, machine learning, with human augmentation requires a thorough strategy to achieve a breakthrough.

The concept of a perception neural network was first described as early as the 1950s. However, it wasn't until recently that the necessary training data, neural-network frameworks, and the requisite processing power came together to help launch an artificial-intelligence (AI) revolution. Despite the tremendous growth of AI technology, the AI revolution continuously requires new tools and methods to take full advantage of its promise, especially when dealing with imaging data beyond visible wavelengths of the electromagnetic spectrum.

One such data type is thermal imaging, or the ability to capture long-wave infrared (LWIR) data. Thermal is a subtype of a much larger world of imaging that emerged in the latter half of the 20th century, including LiDAR and radar.

Thermal imaging has proven critical for many applications. Specific properties make it complementary to other image modalities, as it provides both humans and machines the ability to "see" at night, through smoke and fog, and it's unaffected by sun glare.

Today, organizations are focusing on extracting automated decision support from infrared imagers. Then they combine that data with visible video cameras, radar, and LiDAR, which are deployed in a wide range of systems including automotive safety, autonomy, defense, marine navigation, security, and industrial inspection.

However, leveraging the power of thermal imaging for AI through machine learning, with the help of human augmentation, is easier said than done. Critical to the task is ensuring captured data can be processed closest to each camera. Thus, deep learning and its attendant processing should happen at the edge.

Engineering Focus: Considerations for AI, Compute, and Hardware

There are several important considerations when evaluating AI computing platforms for thermal imaging, The first is the effective number of arithmetic logic units (ALUs) available to perform AI workloads. It's common to utilize neural-network accelerators, GPUs, DSPs, and CPUs for portions of the workload.

When making a hardware selection, it's key to understand the strengths and weaknesses of each platform, and to budget resources accurately. The ability to run multiple software routines simultaneously is critical to automatic target recognition software. Without adequate processing power, the software must fit the various routines into the time slice available, which results in dropped frames.

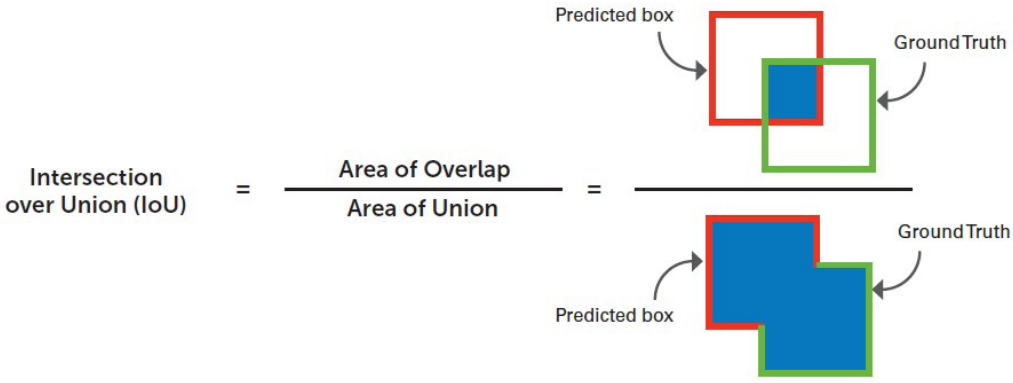
A second consideration is the type and amount of memory the processor can access. Fast and sufficient memory is vital to achieve inference at high frame rates while running all of the software routines, such as warp perspective, optical flow, object tracking, and the object detectors.

A fitting example involves processors featuring integrated LPDDR5 memory with at least 8 GB designed into several intelligent cameras. For example, one widely used type of AI stack software requires between 1 to 5 W when running on an Ambarella CV-2 or Qualcomm RB5165 processor.

Power consumption is managed by selecting networks that fit the power budget specified by the integrator and still meet performance requirements. Object-detection performance is impacted by these configurations, but performance gains continue to be made with more efficient neural networks and new node generation vision-processor hardware.

Processor Choices

Product developers make many decisions when designing cameras with onboard intelligence at the edge. The most impactful is the selection of the vision processor. NVIDIA is a leader in AI compute platforms due to graphics processing technology being ideal for the highly parallel computational demands of neural networks, and for their ecosystem of open-source intellectual property (IP) for network training and runtime deployment.



1. The intersection-over-union (IoU) model can help identify performance measures. (Credit: Teledyne FLIR)

While NVIDIA's platforms continue to be powerful, other leading vision processor suppliers, such as Altera, Ambarella, Intel, MediaTek, NXP, Qualcomm, and Xilinx also have developed highly competitive chip architectures that feature neural-network cores or computational fabrics designed to process neural-network computational loads at significantly lower power and cost.

A growing number of suppliers offer powerful vision processors. However, it's often not feasible for smaller developers to source cutting-edge processors directly from the manufacturers who typically direct smaller volume customers to partner firms offering system-on-module (SOM) solutions and technical support. It can be advantageous to work directly with the vision processor supplier, because the integration of complex multi-threaded runtime routines requires close support from them.

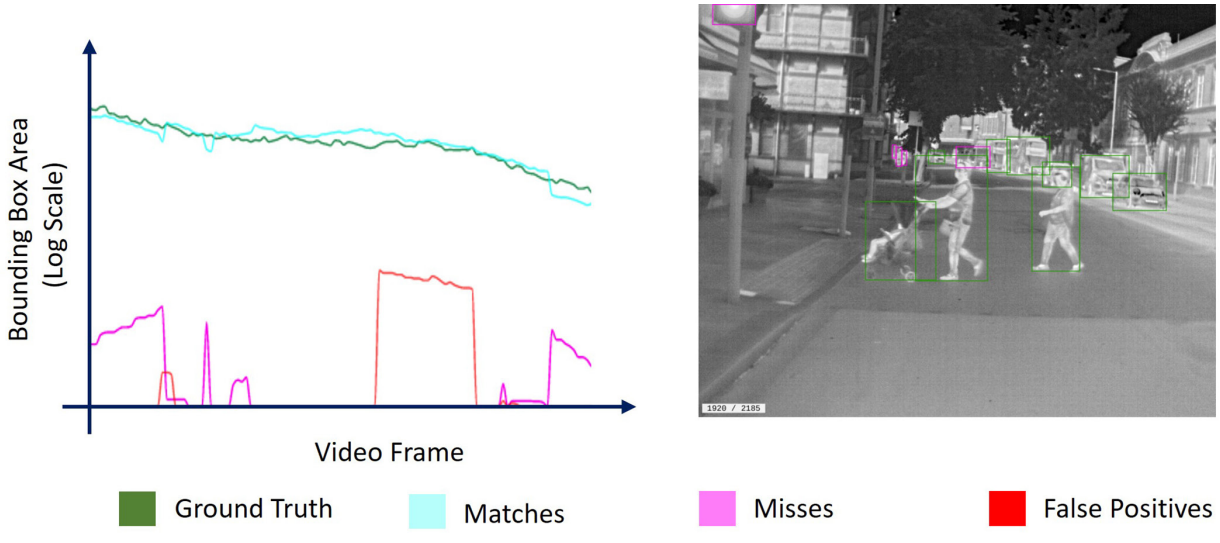
Performance for Convolutional Neural Networks (CNNs)

While an increased number of processor choices exist for running models at the edge, model training is typically done on NVIDIA hardware due to its mature deep-learning development environment built around the company's GPUs.

Neural-network training is computationally demanding, to extremes, and when training a model from scratch, a developer can expect training times of up to five days on a high-end multi-GPU machine. Local servers are typically used for cost considerations, although training also can be performed on popular cloud services.

The second decision a developer must make is to select the neural-network architecture. In the context of computer vision, a neural network is typically defined by its input resolution, operation types, and configuration/number of layers. These factors all translate to the number of trainable parameters that have a high influence on the computational demand. Computational demands translate directly to power consumption and the thermal loads that must be accounted for during the design of products.

The trade space dictates tradeoffs between object-detection accuracy and frame rates for a given vision processor's computational bandwidth. Video camera users



2. This image is an example of visual model performance software output and its associated infrared image. (Credit: Teledyne FLIR)

Model	Input Resolution	Number of Trainable Model Parameters (M)	Multiply-Accumulators (MACs) (GFLOPS)
Inception_v3	300 x 300	27.16	5.75
Resnet50	224 x 224	25.56	4.14
FLIR Compact	480 x 480	8.51	2.41

The table includes neural-network parameters, input resolution, and the associated processing demands for four example models for informational and comparison purposes.

typically demand fast and accurate object detection that enables both human and automatic feedback response by motion-control systems or alarms.

A good example is automatic emergency braking (AEB) for passenger vehicles: A vision-based system detects a pedestrian or other objects within milliseconds and then initiates braking to stop the vehicle, leveraging data from multiple sensor types including radar, visible, and now thermal imaging.

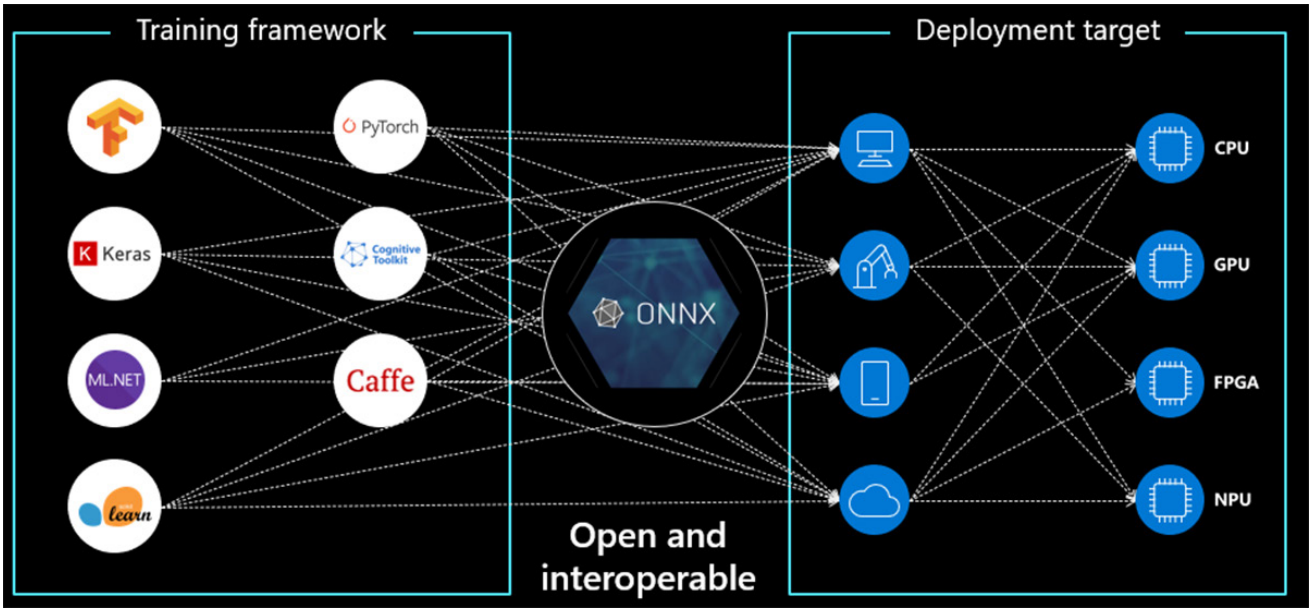
In addition to understanding the performance of models *(Fig. 1)*, data scientists need to analyze the cause of false positives and false negatives or "misses." Here, a subscription-based dataset management cloud software tool can be ideal, especially one that includes a local visual model-performance tool capable of visualizing model performance.

Such a tool can be used to interactively explore and identify areas where the model performs poorly, enabling the data scientist to more effectively investigate the specific training dataset images that cause the missed detections. Developers can then quickly modify or augment the training data, retrain, retest, and iterate until the model converges on the required performance.

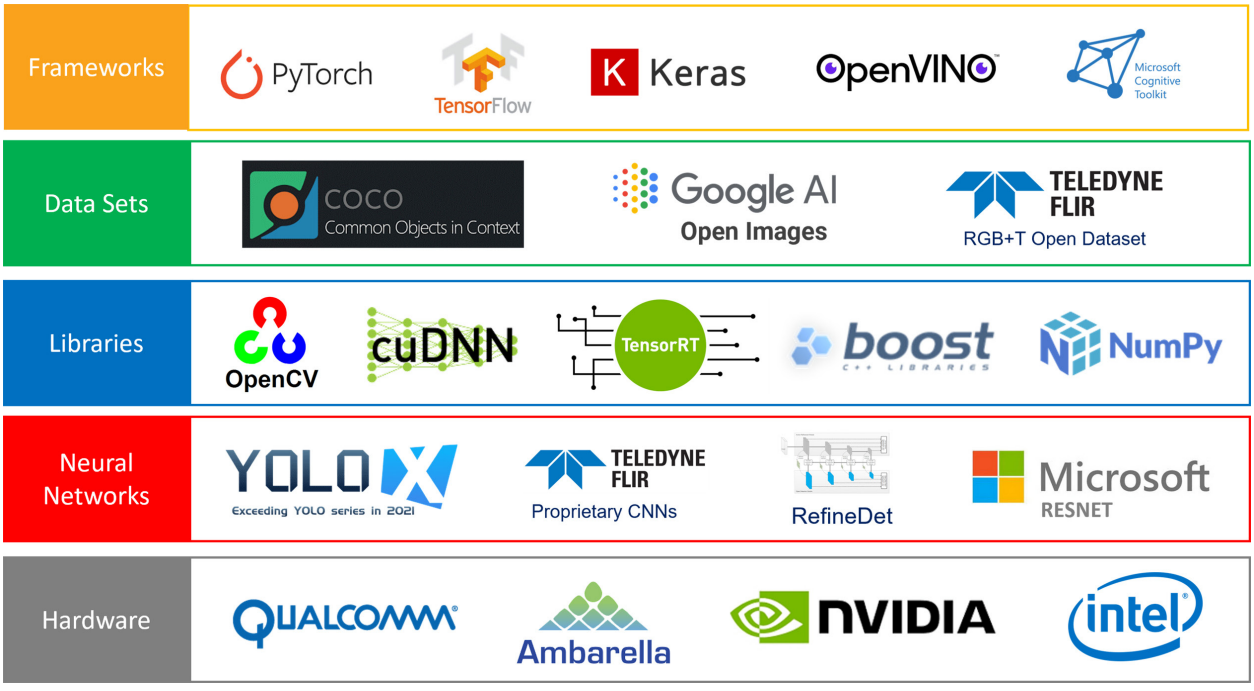
It's also instructive to understand the number of calculations performed by neural networks. For video applications such as automotive safety systems, computations are performed on every video frame *(Fig. 2)*. It's critical to get rapid object detections to eliminate latency. For other applications, including counter unmanned aerial systems (C-UAS), quick detection and object location meta data is a critical input into a video tracker that controls the camera and counter-measure pointing actuators.

The *table* includes neural-network parameters, input resolution, and the associated processing demands for four example models for informational and comparison purposes. These estimations don't account for how well the architecture utilizes the specific hardware, so it's important to note that the most reliable way to benchmark a model is to run the model on the actual device.

At the end of a training process, the model typically needs to be converted to run on the target vision processor's specific execution fabric. The translation and fit process is extremely complex and requires a skilled software engineer. This has been a significant point of friction in preventing a faster



3. This image depicts the OXXN AI open-source operational model to help standardize a model file format and associated tools. (Credit: https://microsoft.github.io/ai-at-edge/docs/onnx/)



4. Shown are notable organizations across the entire AI stack. (Credit: Teledyne FLIR)

deployment of AI at the edge.

In response, an industry consortium established ONNX AI, an open-source project that established a model file-format standard and tools to facilitate runtime on a wide range of processor targets *(Fig. 3)*. As ONNX becomes fully supported by vision-processor suppliers and the developer community, the efforts required to deploy models on different hardware will significantly reduce a pain point for developers.

AI Stack Development

Figure 4 includes examples of the frameworks, datasets, libraries, neural networks, and hardware that make up the typical AI stack. Vendors continue to develop and manufacture LWIR, mid-wave infrared (MWIR), and visible light cameras that can and are being developed to utilize AI at the edge. Given the requirements and lack of mature tools associated with multispectral sensing, in particular, unique software, datasets, and more must be developed to support AI at the edge for different sensor types.

To achieve this, a PyTorch framework can be implemented, which is tightly integrated with Python. PyTorch supports dynamic computational graphs allowing the network behavior to be changed programmatically at runtime. In addition, the data parallelism feature allows PyTorch to distribute computational work among multiple GPUs as well as multiple machines to decrease training time and improve accuracy.

Datasets for object detection are large collections of images, whether or not they consist of thermal or visible images, that have been annotated and curated for class balance and characteristics such as contrast, focus, and perspective. It is industry best practice to manage a dataset like software source code and utilize revision control to track changes. This ensures machine-learning models maintain consistent and reproducible performance.

If a developer encounters performance issues with a model, data scientists can quickly identify where to augment the dataset to create a continuous improvement lifecycle. Once a verified improvement is made, the data change is recorded with a commit entry that can then be reviewed and audited.

While open-source datasets like Common Objects in Context (COCO) are available, they're visible light image collections containing common objects at close range captured from a ground-level perspective. Meanwhile, many emerging applications, as mentioned above, are now requiring multispectral images taken from a variety of angles and contexts, from air to ground, ground to air, across water, and of unique objects, including military objects.

For example, to facilitate the evaluation of thermal imaging for automotive safety and autonomy systems developers, open-source datasets can show matched thermal and visible frames. This way, images of targets at various distances can be added to ensure models work well as small object detectors.

Incorporating Synthetic Data to Improve CNN Efficiency and Performance

In the real world, objects are viewed in near-infinite combinations of distance, perspective, background environments, and weather conditions. The accuracy of machine-learning models largely depends on how well training data represents field conditions.

Modern tools that analyze a dataset's imagery and quantify the data distribution based on object label (e.g., the percent of images of person, car, bicycle, etc.), object size, contrast, sharpness, and brightness are extremely helpful. Correlating model performance to data characteristics and then producing a datasheet for each new model release is valuable to data scientists and critical in the ongoing iteration of model development.

The challenge for data scientists is the significant time and expense requirements to build large training datasets, requiring field data collection, curation of frames, annotation, and quality control of label accuracy. This is a bottleneck in deploying AI, but advances in synthetic data can help reduce development time.

5. Here's an example of synthetic training data of a tank in thermal imaging and in visible light. (Credit: CVEDIA)

Synthetic data helps to create multispectral data and models using computer-generated imagery (CGI), allowing for the creation of multispectral imagery of almost any object from any perspective and distance *(Fig. 5)*. The result is the ability to create datasets of unique objects like foreign military vehicles, or in relatively rare weather environments, such as heavy fog, that would be extremely challenging to do when relying on field data collection only.

AI at the Edge in Production

There's a convergence of development and technology enabling a clearer path to deploy affordable and functional AI at the edge. Lower-cost hardware is being released with improved processing performance that can be used with more efficient neural networks.

Software tools and standards to simplify model creation and deployment are promising and ensure developers can add AI to their respective cameras with lower monetary investment. The open-source community and model standards from the ONNX community are contributing to this while also aiding in the acceleration of AI at the edge.

As integrators demand AI at the edge in industrial, automotive, defense, marine, security, and other markets, it's important to recognize the engineering effort required to move a proof-of-concept demonstration of AI at the edge to production.

Developing training datasets, addressing performance gaps, updating training data and models, and integrating new processors requires a team with diverse skills. Imaging-system developers will need to carefully consider the investment required to build this capability internally or when selecting suppliers to support their respective AI stacks.

Art Stout is charged with leading the Teledyne FLIR development of advanced deep learning, artificial intelligence, and machine learning products at Teledyne FLIR, including models, datasets, dataset management software, and AI services. He brings 30 years of imaging industry experience serving in a variety of roles in sales management, marketing, business development, product management, and strategic corporate development.