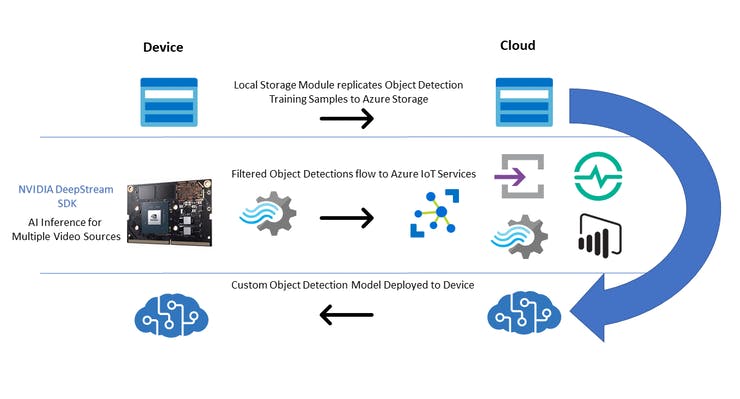
**Introduction**

With the advent of Artificial Intelligence and Internet of Things, a new paradigm of AIOT solutions is beginning to emerge. This is in part due to hardware advancements that allow for accelerated workloads to run on small form-factor edge devices in addition to software development kits that are targeted to these devices and AI use cases. In this post, we will specifically look at the [NVIDIA Jetson family of devices](https://developer.nvidia.com/buy-jetson) and the [NVIDIA DeepStream SDK](https://developer.nvidia.com/deepstream-sdk), a platform that allows for optimized deployment of accelerated AI workloads to a device not much larger than a cell phone.

**Video Analytics at the Edge**

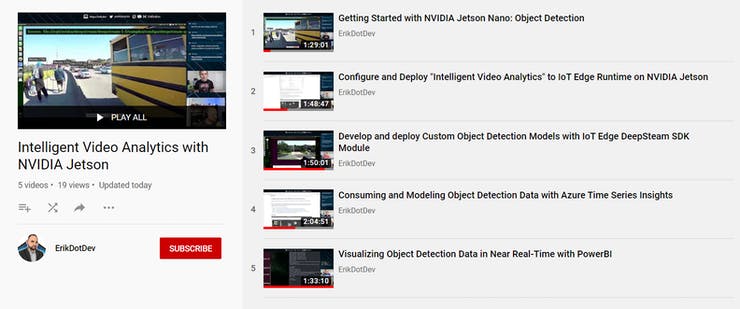
Video sources can be used in combination with Artificial Intelligence to perform a variety of useful tasks. These could include [anomaly detection in manufacturing scenarios](https://developer.nvidia.com/gtc/2020/video/s22675-vid), [self-driving vehicles](https://www.youtube.com/watch?v=syphzPxnozA), or even [sorting Lego pieces](https://www.theverge.com/2019/12/11/21011792/lego-ai-universal-sorting-machine). Intelligent Video Analytics solutions require a great deal of cross-domain knowledge in order to implement. For example, you need to optimize the acquisition and decoding of frames for the number of cameras involved, techniques for training, accelerating, and optimizing AI Inference workloads, and an ability to publish inference results out to local and remote reporting services. These problems are difficult, but with the use of tools like the NVIDIA DeepStream SDK, much of these problems are solved for you, allowing you to focus on developing a solution that meets your specific requirements. The diagram below depicts the solution we will be developing in this article, take note of the NVIDIA Jetson hardware and inclusion of the DeepStream SDK and Azure Services for reporting.



Intelligent Video Analytics architecture

**Getting Started**

To demonstrate how to create an Intelligent Video Analytics solution, as part of [#JulyOT](https://www.hackster.io/news/julyot-a-month-dedicated-to-learning-and-building-iot-projects-6408dbc4224a), we have published a [Github repository](https://github.com/toolboc/Intelligent-Video-Analytics-with-NVIDIA-Jetson-and-Microsoft-Azure) of best practices in the form of [video content](https://www.youtube.com/watch?v=yZz-4uOx_Js&list=PLzgEG9tLG-1QLc-DPPABoW1YWFMPNQl4t&index=1) and code templates that can theoretically enable you to build an end-to-end custom object detection system with analytics and reporting to the Azure Cloud. The amazing thing about this content is that our videos were recorded live, while building out the entire solution from scratch with developer [Erik St. Martin](https://twitter.com/erikstmartin). The important thing to note here is that all of the topics that we will cover were brand new to him and may very well be completely new to you too! This has allowed us a unique opportunity to distill all of the various intricacies involved in developing a custom Intelligent Video Analytics solution into bite-sized chunks, resulting in approximately 8 hours of instructional content designed to teach anyone how to build their own solution!



~8 hours of recorded video to teach you how to build an Intelligent Video Analytics solution

To begin, you will want to head to the [Github repository](https://github.com/toolboc/Intelligent-Video-Analytics-with-NVIDIA-Jetson-and-Microsoft-Azure), then you can head back here to go over each of the modules contained within, with the benefit of some additional background info on the objectives of each module.

**Module 1 - Introduction to NVIDIA DeepStream**

The [NVIDIA DeepStream SDK](https://developer.nvidia.com/deepstream-sdk)delivers a complete streaming analytics toolkit for AI based video and image understanding and multi-sensor processing. DeepStream SDK features hardware-accelerated building blocks, called plugins that bring deep neural networks and other complex processing tasks into a stream processing pipeline.

The deepstream offering contains the DeepStream SDK which include an app (deepstream-test5) that is configurable to handle multiple streams and multiple networks for inference. The app can be connected to the [Azure IoT Edge runtime](https://docs.microsoft.com/en-us/azure/iot-edge/about-iot-edg?WT.mc_id=julyot-iva-pdecarlo)to send messages to a configured [Azure IoT Hub.](https://docs.microsoft.com/en-us/azure/iot-hub/?WT.mc_id=julyot-iva-pdecarlo)

The [DeepStream SDK is offered in the Azure Marketplace](https://azuremarketplace.microsoft.com/en-us/marketplace/apps/nvidia.deepstream-iot?WT.mc_id=julyot-iva-pdecarlo)as an [IoT Edge Module.](https://docs.microsoft.com/en-us/azure/iot-edge/quickstart-linux?WT.mc_id=julyot-iva-pdecarlo) We will employ this mechanism to configure and run a DeepStream workload on an NVIDIA embedded device.

Before continuing, it is highly suggested to familiarize with the [DeepStream SDK Documentation,](http://aka.ms/deepstreamdevguide) as it will provide you with the details on how to customize the Intelligent Video Analytics solution to your needs.

We cover pretty much everything you need to know in this 90 minute livestream titled "[Getting Started with NVIDIA Jetson: Object Detection"](https://www.youtube.com/watch?v=yZz-4uOx_Js). We highly recommend that you give a watch before proceeding to the next section.

Module 1 - Introduction to NVIDIA DeepStream

**Module 2 - Configure and Deploy "Intelligent Video Analytics" to IoT Edge Runtime on NVIDIA Jetson**

In this section we will install and configure the [IoT Edge Runtime](https://docs.microsoft.com/en-us/azure/iot-edge/about-iot-edge?WT.mc_id=julyot-iva-pdecarlo)on an NVIDIA Jetson Device. This will require that we deploy a collection of Azure Services to support the modules that are defined in the associated [IoT Edge Deployment for IoT Hub.](https://github.com/toolboc/Intelligent-Video-Analytics-with-NVIDIA-Jetson-and-Microsoft-Azure/blob/master/deployment-iothub/deployment.template.json)

In this section, we will only need to deploy an [Azure IoT Hub](https://docs.microsoft.com/en-us/azure/iot-hub/?WT.mc_id=julyot-iva-pdecarlo)and [Azure Storage Account.](https://docs.microsoft.com/en-us/azure/storage/?WT.mc_id=julyot-iva-pdecarlo) If you are curious about the pricing involved for these services, they are summarized below:

* [IoT Hub Pricing](https://azure.microsoft.com/en-us/pricing/details/iot-hub/?WT.mc_id=julyot-iva-pdecarlo)
* [Azure Storage Account](https://github.com/toolboc/Intelligent-Video-Analytics-with-NVIDIA-Jetson-and-Microsoft-Azure?WT.mc_id=julyot-iva-pdecarlo)
* [Azure Stream Analytics on Edge Pricing](https://azure.microsoft.com/en-us/pricing/details/stream-analytics/?WT.mc_id=julyot-iva-pdecarlo)(Technically, even though we are using a job that is not contained in the end-users subscription, billing does occur per device that runs the DeepStreamAnalytics Module)

The additional services, [CustomVision.AI](https://www.customvision.ai/?WT.mc_id=julyot-iva-pdecarlo)and [Azure Stream Analytics on Edge,](https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-edge?WT.mc_id=julyot-iva-pdecarlo) will be addressed in upcoming sections and will not be needed at this time.

If you wish to follow along with the steps in this module, we have recorded a livestream presentation titled "[Configure and Deploy "Intelligent Video Analytics" to IoT Edge Runtime on NVIDIA Jetson"](https://www.youtube.com/watch?v=RKwwP4XsZdw) that walks through the steps below in great detail.

Module 2 - Configure and Deploy "Intelligent Video Analytics" to IoT Edge Runtime on NVIDIA Jetson

**Module 3 - Develop and deploy Custom Object Detection Models with IoT Edge DeepStream SDK Module**

At this point, you should have deployed a custom DeepStream Configuration that is able to consume input from your desired sources. We will now look into ways to customize the object detection model that is employed in that configuration to enable to you to create a fully customized Intelligent Video Analytics Pipeline.

This section will assume that you might be brand new to the world of Computer Vision / Artificial Intelligence and that you are interested in the end goal of using a Custom Object Detection model that detects objects that you train it to detect. If you are interested in obtaining accurate detection of common objects immediately, without the need to train up a custom model, we will also demonstrate how to employ an academic-grade pre-trained object detection model [(YoloV3)](https://pjreddie.com/darknet/yolo/)which has been trained on [80 common objects.](https://github.com/toolboc/Intelligent-Video-Analytics-with-NVIDIA-Jetson-and-Microsoft-Azure/blob/master/services/YOLOV3/labels.txt)

If you wish to follow along with the steps in this module, we have recorded a livestream presentation titled "[Develop and deploy Custom Object Detection Models with IoT Edge DeepSteam SDK Module"](https://www.youtube.com/watch?v=kv0eTobemug) that walks through the steps below in great detail.

Module 3 - Develop and deploy Custom Object Detection Models with DeepStream SDK Module

**Module 4 - Filtering Telemetry with Azure Stream Analytics at the Edge and Modeling with Azure Time Series Insights**

At this point, you should have a working DeepStream Configuration referenced by the NVIDIADeepStreamSDK module, [customized to accommodate your video input source(s),](https://github.com/toolboc/Intelligent-Video-Analytics-with-NVIDIA-Jetson-and-Microsoft-Azure/blob/master/docs/Module%202%20-%20Configure%20and%20Deploy%20Intelligent%20Video%20Analytics%20to%20IoT%20Edge%20Runtime%20on%20NVIDIA%20Jetson.md#module-26--customizing-the-sample-deployment) and [configured to use a custom object detection model.](https://github.com/toolboc/Intelligent-Video-Analytics-with-NVIDIA-Jetson-and-Microsoft-Azure/blob/master/docs/Module%203%20-%20Develop%20and%20deploy%20Custom%20Object%20Detection%20Models%20with%20IoT%20Edge%20DeepSteam%20SDK%20Module.md)

In this module we will explain how to flatten, aggregate, and summarize DeepStream object detection results using [Azure Stream Analytics on Edge](https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-edge?WT.mc_id=julyot-iva-pdecarlo)and forward that telemetry to our [Azure IoT Hub.](https://docs.microsoft.com/en-us/azure/iot-hub/?WT.mc_id=julyot-iva-pdecarlo) We will then introduce a new Azure Service known as [Time Series Insights.](https://docs.microsoft.com/en-us/azure/time-series-insights/?WT.mc_id=julyot-iva-pdecarlo) This service will take in input via an event-source from our IoT Hub to allow us to analyze, query, and detect anomalies within the object detection data produced by our IoT Edge device.

If you wish to follow along with the steps in this module, we have recorded a livestream presentation titled "[Consuming and Modeling Object Detection Data with Azure Time Series Insights"](https://www.youtube.com/watch?v=l5tjaD-qYwY) that walks through the steps below in great detail.

Module 4 - Filtering Telemetry with Azure Stream Analytics and Modeling with Time Series Insights

**Module 5 - Visualizing Object Detection Data in Near Real-Time with PowerBI**

Power BI is a business analytics service provided by Microsoft. It provides interactive visualizations with self-service business intelligence capabilities, where end users can create reports and dashboards by themselves, without having to depend on information technology staff or database administrators.

In this module, we will cover how to forward object detection telemetry from our Azure IoT Hub into a PowerBI dataset using a cloud-based Azure Stream Analytics job. This will allow us to build a report that can be refreshed to update as detections are produced. We will then Publish a PowerBI report and convert it to a live dashboard. From there, we can query our data with natural language and interact with our data in near real-time.

In order to complete this module, it will require that you have an active PowerBI account. If you need to create one, this [video](https://channel9.msdn.com/Blogs/BretStateham/Signing-up-for-Power-BI)walks through the process.

If you wish to follow along with the steps in this module, we have recorded a livestream presentation titled "[Visualizing Object Detection Data in Near Real-Time with PowerBI"](https://www.youtube.com/watch?v=lhvPbNF9eb4) that walks through the steps below in great detail.

Module 5 - Visualizing Object Detection Data in Near Real-Time with PowerBI

**Conclusion**

At this point, assuming you have gone through all of the included materials, you now know how to develop DeepStream applications using a variety of video input sources (USB Camera / RTSP / Looping File), how to containerize a DeepStream Workload for deployment as an IoT Edge module, utilize various services to gather samples, train, and deploy a custom object detection model, and how to publish results into Cloud-based services like Azure Time Series Insights and PowerBI.  
  
This is a HUGE accomplishment, and likely a very employable skillset at this time. The ~8 hour time investment for learning these techniques is necessary to fully demonstrate all of the components that make up an Intelligent Video Analytics service. It is our hope that you have found this content valuable and able to apply it to your specific scenario.