Unlocking Vision AI:
From Edge to Insight with STMicroelectronics & Avnet

May 2025

Agenda & Speakers

- **Welcome – Keaton Andersen**
- **Introduction to Vision AI – Keaton Andersen**
- **ST Bright Sense Camera Sensors – Thomas Viart**
- **Optimizing Camera Sensors from the Cloud – Keaton**
- **Vision AI Models and Use Cases – Steven Dettloff**
  - Hardware Overview
  - Image Classification
  - Object Detection and People Detection
  - Instance and Semantic Segmentation
  - Pose Estimation
- **Q&A + Resources - All**

Keaton Andersen
Sr Manager, Customer Solutions
Avnet

Thomas Viart
Imaging Regional Marketing
STMicroelectronics

Steven Dettloff
IoT Solutions Manager
Avnet

VisionAI Applications Featuring STM32N6

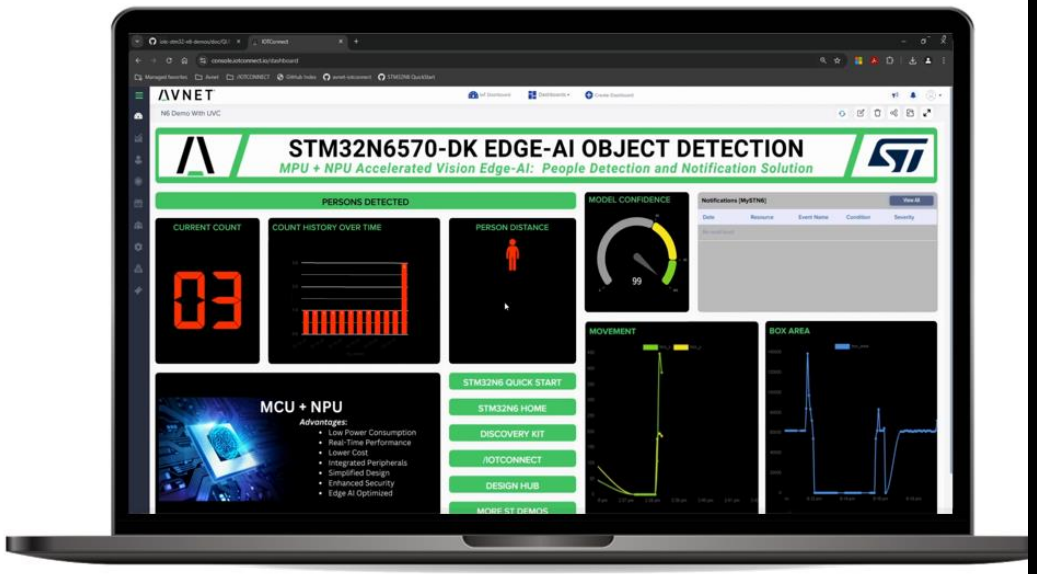
- Overview
- Discovery Kit with STM32N6
- /IOTCONNECT
- ST Bright Sense Camera Modules

VisionAI Models
(YoloV8, ResNet, etc)

Attached Camera Module

Discovery Kit with
STM32N657X0 MCU

/IOTCONNECT™

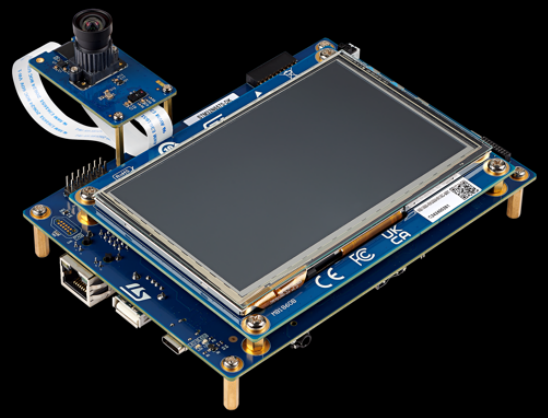


WiFi Added by
DA16200 Pmod™

/IOTCONNECT™
powered by aws

VisionAI Applications Featuring STM32N6

- Overview
- Discovery Kit with STM32N6
- /IOTCONNECT
- ST Bright Sense Camera Modules



Discovery Kit with
STM32N657X0 MCU

Embedded memories
- 4.2 MB Contiguous RAM
- BOOT 128KB
- 8 kB Backup RAM
- 4 kB Debug RAM

External memory interfaces
- Hexa-SPI
- Octo-SPI
- 2x SD/SDIO /MMC
- Flexible Memory Controller (PSRAM, SDRAM, NOR, NAND, TFT-LCD)

Camera pipeline
- MIPI CSI-2, 2 lanes Camera I/F Parallel 16b With PSSI
- Image Signal Processor

Graphics & multimedia
- Chrom-ART Accelerator
- H.264 Encoder
- Chrom-GRC
- JPEG hardware accelerator
- NeoChrom Accelerator
- LTDC/TFT-LCD Display Controller

Arm® Cortex®-M55 800 MHz
TrustZone®
MVE

ST Neural-ART accelerator 600 Gops

System
- 96-bit unique ID
- Watchdogs
- Cyclic Redundancy Check (CRC)
- XTAL & Internal osc.

Security
- Secure boot
- Secure RTC
- Analog true RNG
- Crypto/Hash
- Tampering
- OTP Fuses 8 KB

Connectivity
- 3x CAN FD
- 2x USB 2.0 D/H HS + HS PHY
- 1x UCPD controller
- 3x I²C + 2x I3C (SDR)
- 5x UART, 5x USART
- 6x SPI (incl. 4x I2S)
- 1x ULP UART
- 1x Gbit Ethernet with TSN

Audio
- SPDIF-RX/TX
- 2x SAI
- 1x MDF (6 Filters)
- 1X ADF filter (with SAD)

Analog
- Digital Temperature Sensor
- 2 x 12-bit ADC 5 MSPS

Timers/Control
- 2x 16-bit motor control PWM synchronized AC timer
- 12x 16-bit timer
- 5x 16-bit LP timer
- 4x 32-bit advanced timers

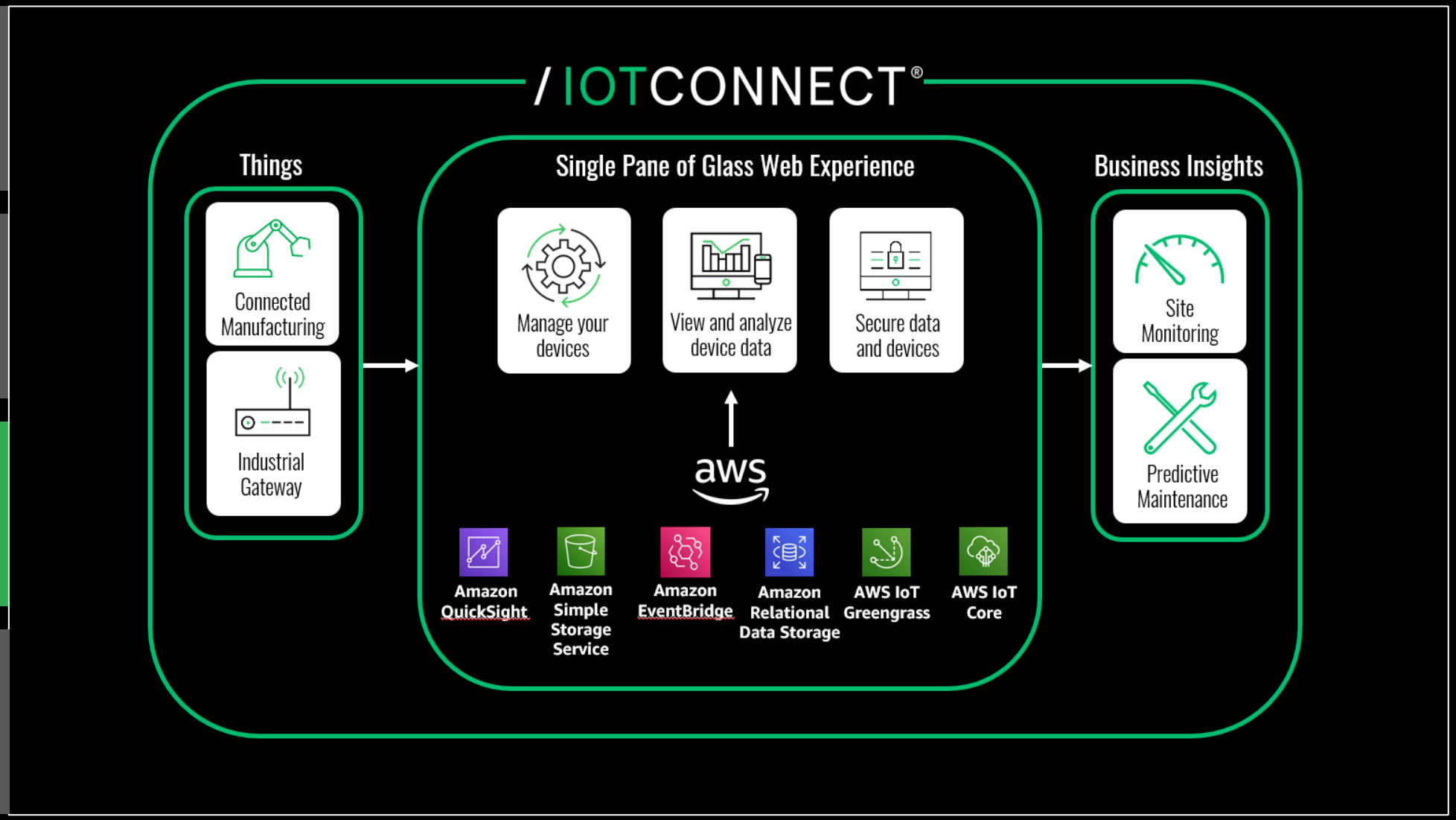
VisionAI Applications Featuring STM32N6

Overview

Discovery Kit with STM32N6

/IOTCONNECT

ST Bright Sense Camera Modules



VisionAI Applications Featuring STM32N6

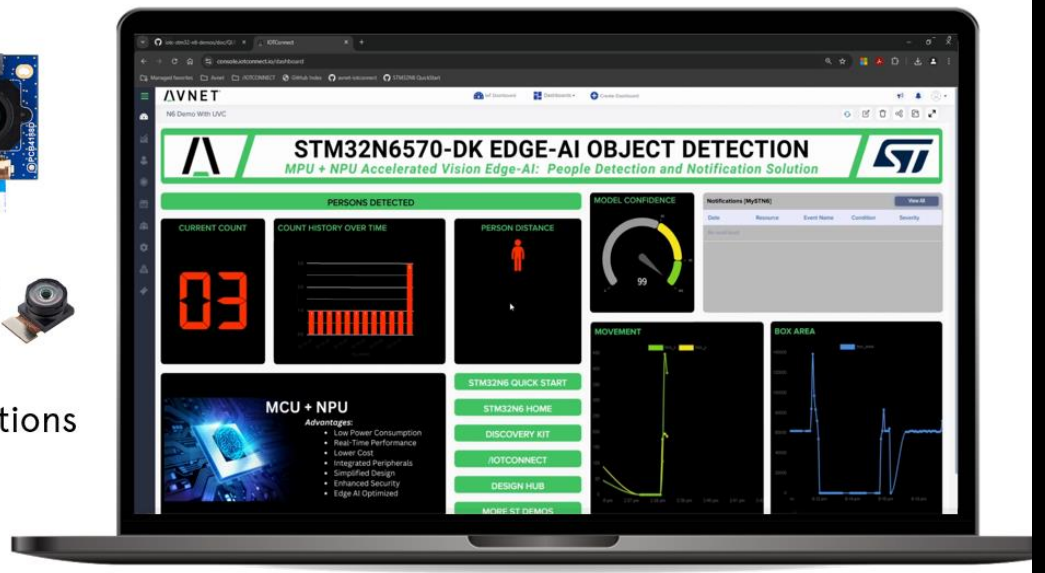
- Overview
- Discovery Kit with STM32N6
- /IOTCONNECT
- ST Bright Sense Camera Modules

VisionAI Models
(YoloV8, ResNet, etc)

Discovery Kit with
STM32N657X0 MCU

ST Camera Options

/IOTCONNECT™



Wifi Added by
DA16200 Pmod™

/IOTCONNECT™
powered by aws

VisionAI Applications Featuring STM32N6

- Overview
- Object Detection Demo
- Instance Segmentation Demo
- Pose Estimation Demo

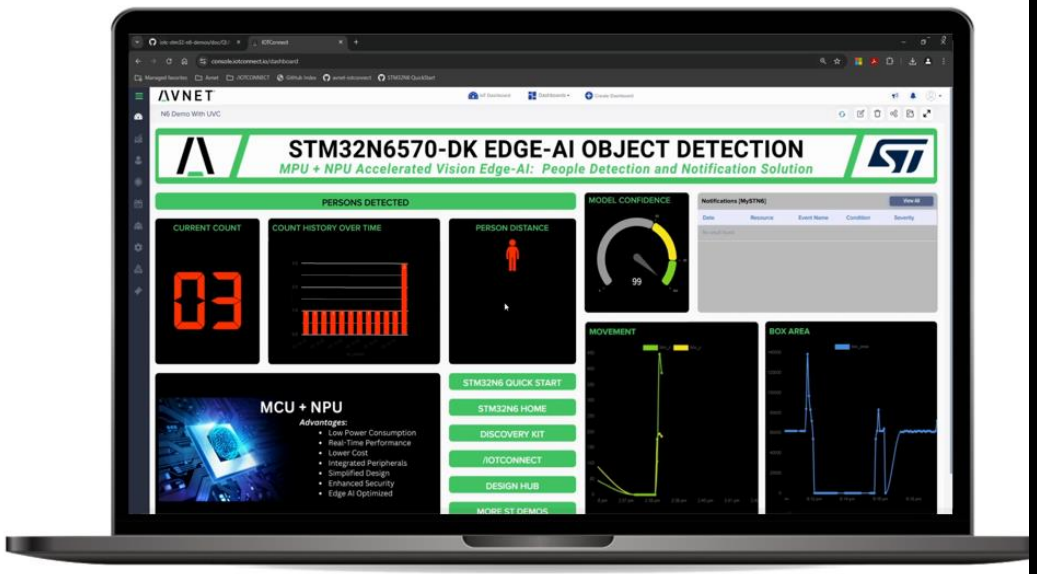
VisionAI Models
(YoloV8, ResNet, etc)

Attached Camera Module

Discovery Kit with
STM32N657X0 MCU

Wifi Added by
DA16200 Pmod™

/IOTCONNECT™

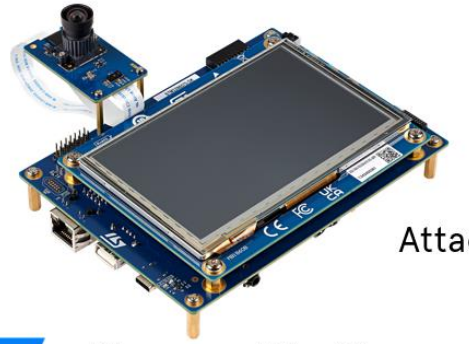


/IOTCONNECT™
powered by aws

VisionAI Applications Featuring STM32N6

- Overview
- Object Detection Demo
- Instance Segmentation Demo
- Pose Estimation Demo

Object Detection Model
(Tiny Yolo v2)



Attached Camera Module

Discovery Kit with
STM32N657X0 MCU

Wifi Added by
DA16200 Pmod™

/IOTCONNECT™ powered by aws

/IOTCONNECT™

VisionAI Applications Featuring STM32N6

- Overview
- Object Detection Demo
- Instance Segmentation Demo
- Pose Estimation Demo

Instance Segmentation Model (Yolov8)

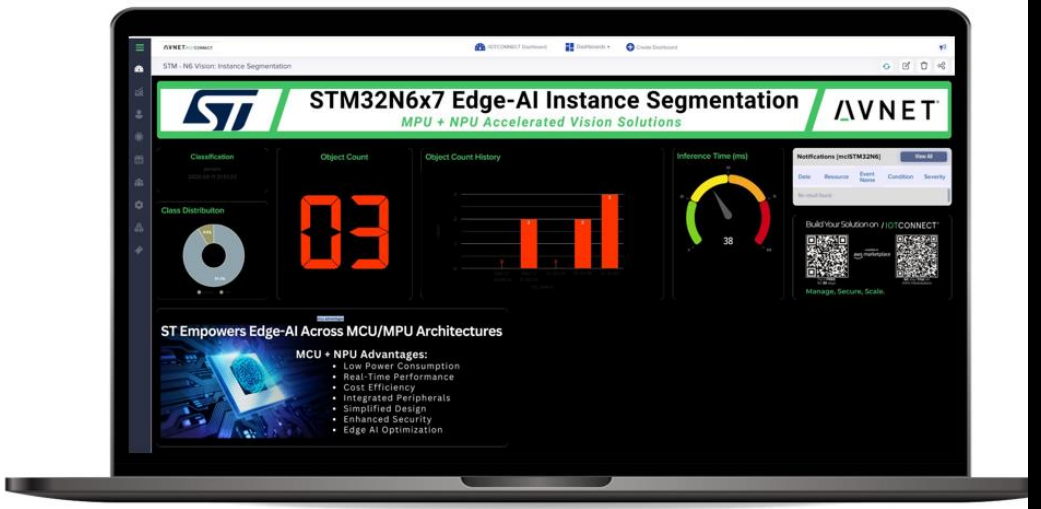
Attached Camera Module

Discovery Kit with STM32N657X0 MCU

Wifi Added by DA16200 Pmod™

/IOTCONNECT™ powered by aws

/IOTCONNECT™



VisionAI Applications Featuring STM32N6

- Overview
- Object Detection Demo
- Instance Segmentation Demo
- Pose Estimation Demo

Pose Estimation Model (MoveNet)

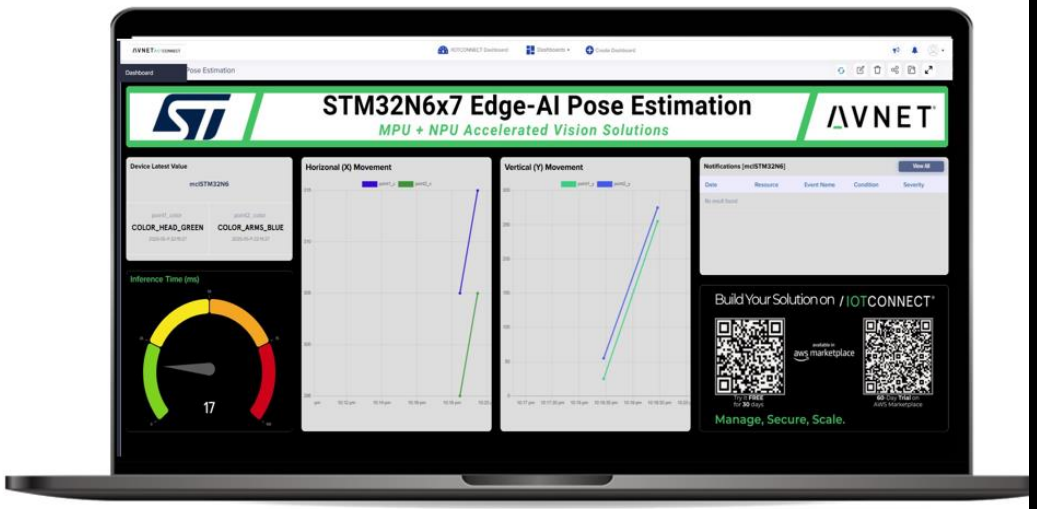
Attached Camera Module

Discovery Kit with STM32N657X0 MCU

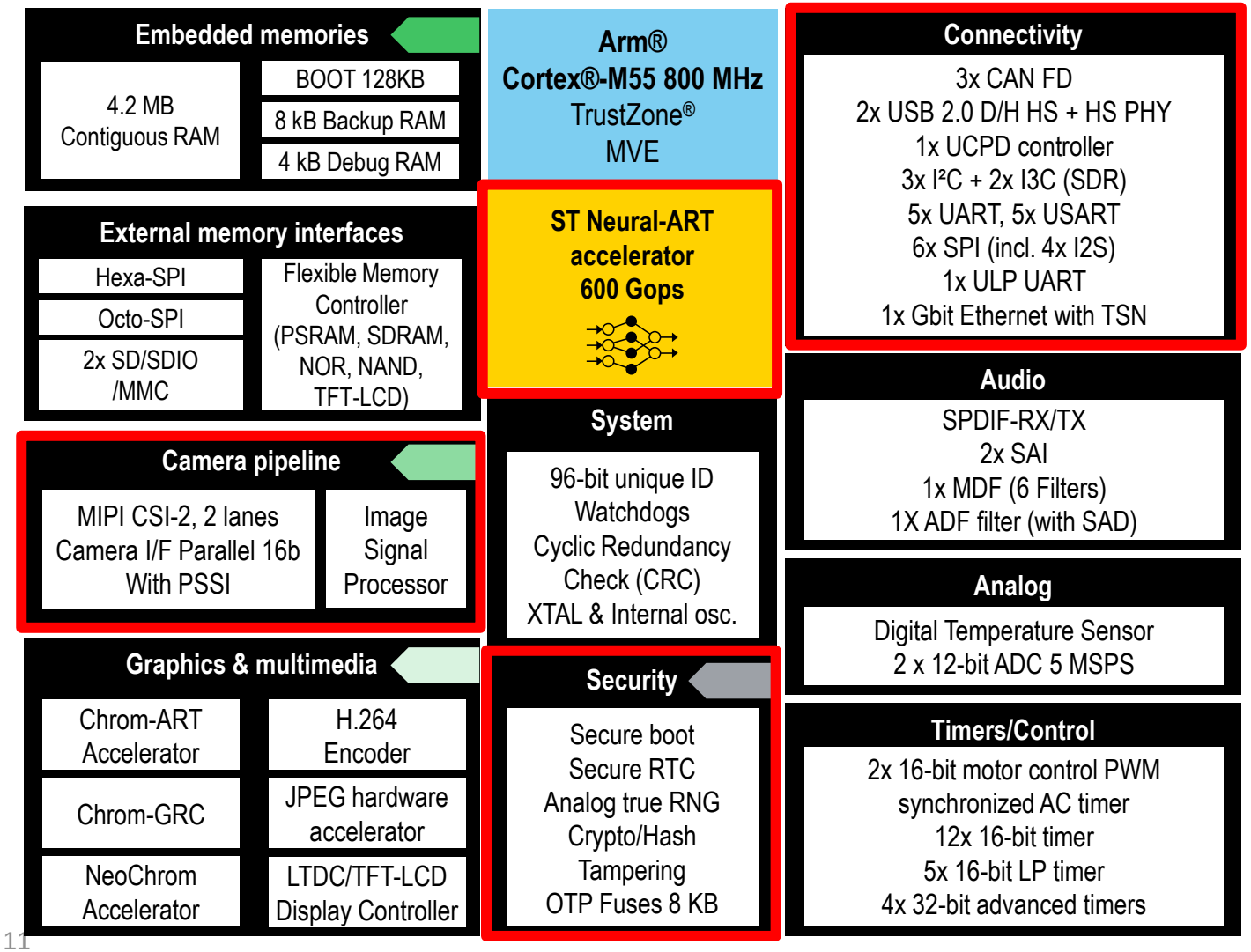
Wifi Added by DA16200 Pmod™

/IOTCONNECT™ powered by aws

/IOTCONNECT™



STMicroelectronics STM32N6x7 MCU with Avnet Support!



Your VALUE-ADD Opportunity!

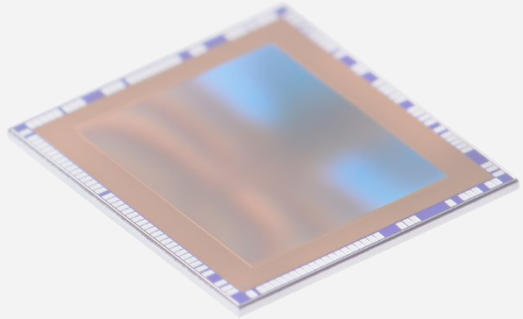
- Connectivity
- Image Sensing
- Applied AI
- Security

STMicroelectronics: the imaging one-stop shop

dToF modules
direct Time-Of-Flight

All-in-one modules for detecting presence and measuring single or multiple distances

iToF sensors
indirect Time-Of-Flight

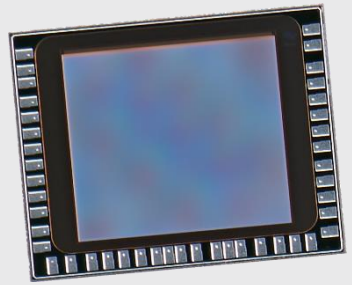


Sensors for accurately capturing 3D depth images over a wide range of distances

ALS sensors
Ambient Light Sensing

Sensors measuring light information to adapt displays, detect colors, or extract flicker frequency

Image sensors
Global shutter

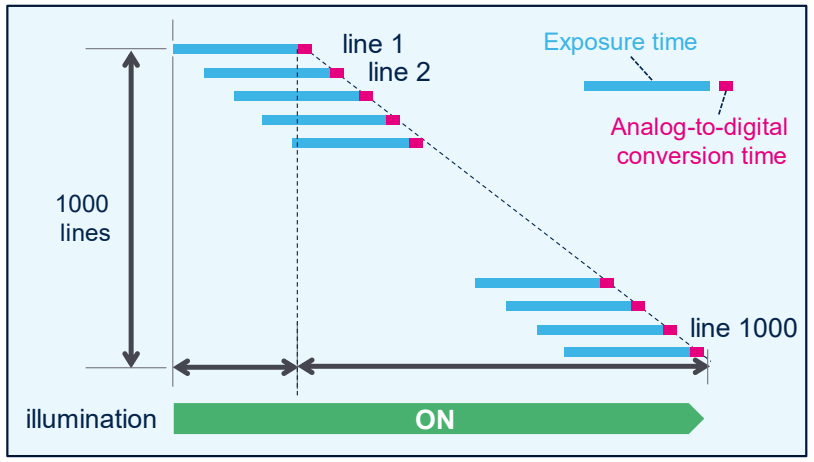


Sensors for capturing images or recording videos, enabling the system to “see”

Two main categories of image sensors: rolling shutter and global shutter

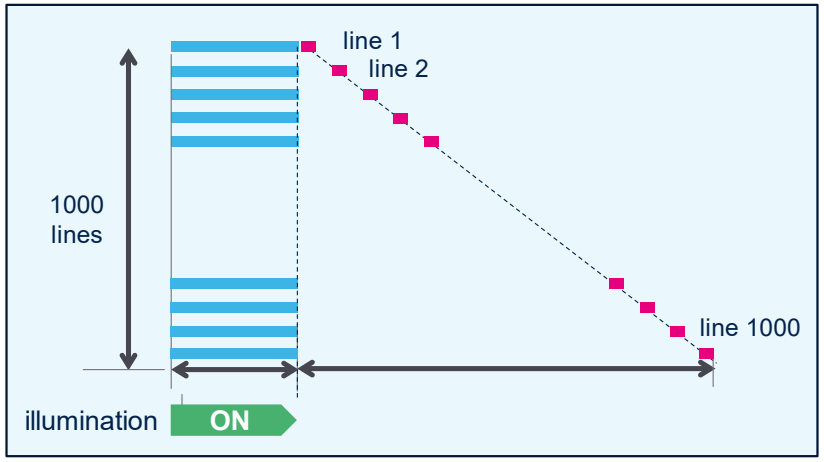
Rolling shutter

Exposure to light is “rolling”: pixel rows are exposed to light one after the other with a delay.



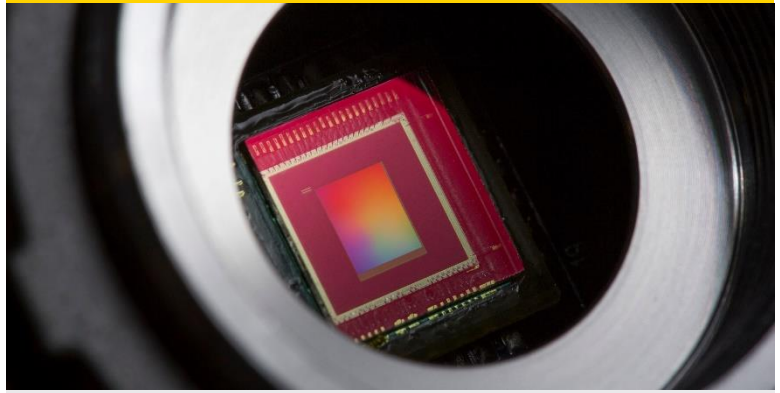
Global shutter

Exposure to light is “global”: pixel rows are all exposed at the same time.



ST BrightSense at a glance

Bring computer vision to the next level



Build up precise and reactive systems
Capture high-quality images in any condition with our advanced pixel technologies and features

Reveal the unseen with Near-InfraRed
Benefit from sensors superior NIR sensitivity for face ID, low-light imaging or bio sign monitoring

Stay at the forefront of innovation
Benefit from state-of-the-art patented technologies from ST's own European wafer fab

Create smart & power-efficient vision solutions

Develop new smart functionalities
Leverage smart embedded features to develop new functions and save MCU resources for other uses

Vision systems to fit in everywhere
Match ultra-compact sensors with your MCU to build tiny and discrete embedded vision systems

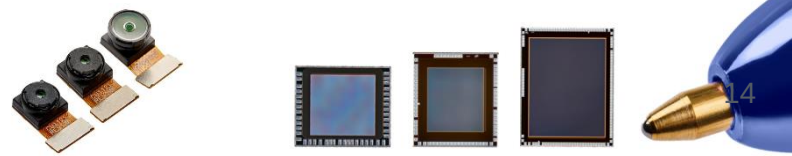
Extend battery life in mobile systems
Leverage low operating power and auto-wake up feature to get rid of inefficient 24/7 operation

Save resources and accelerate your time-to-market

Everything you need in a few clicks
Documentation, reference designs & turnkey hardware to operate with various platforms

Effortless evaluation & development
Use STM32Cube environment with optimized ISP configuration to maximize image quality

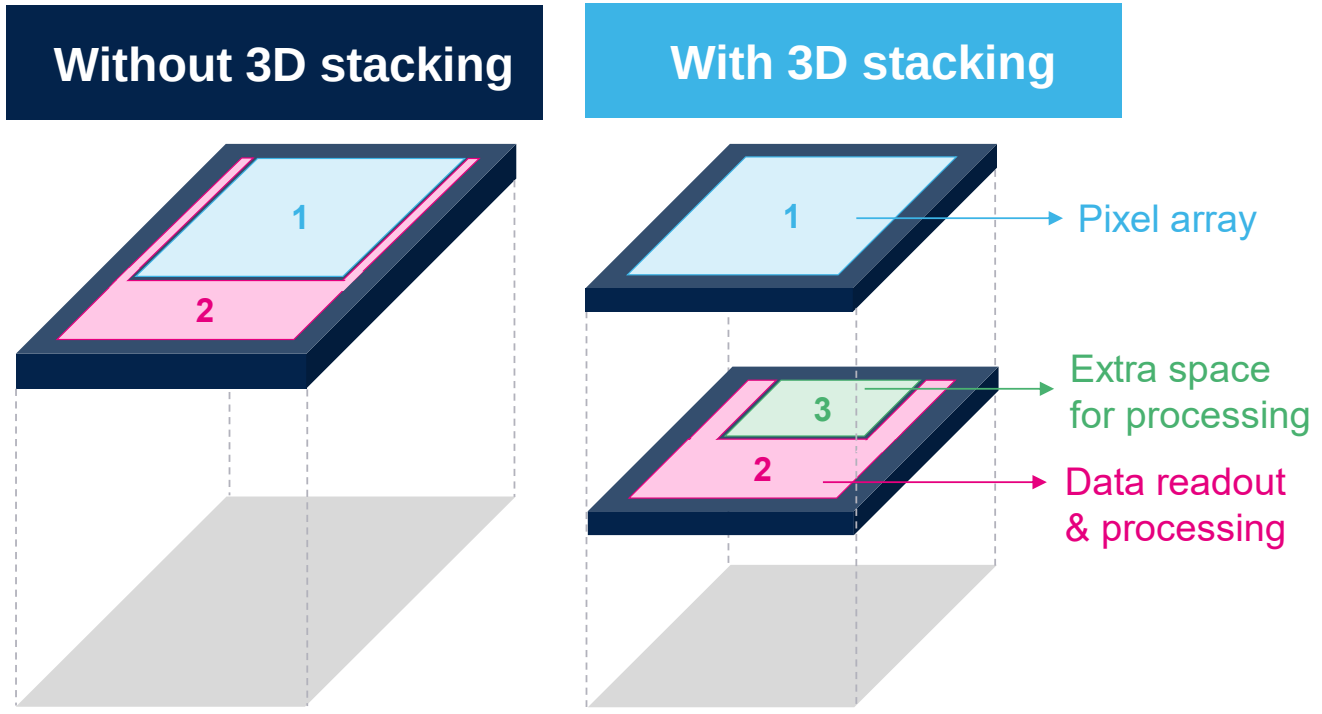
Support from prototype to production
Wide range of evaluation camera modules available then in production from various partners



The benefits of 3D stacking technology

A rich toolbox of on-chip features enabling smarter and low-power system designs

ST BrightSense products feature further **3D-stacking** technology through **advanced image processing features** that free up processor resources for critical tasks and enhanced functionality in a reduced footprint.



With same area surfaces **1** and **2**

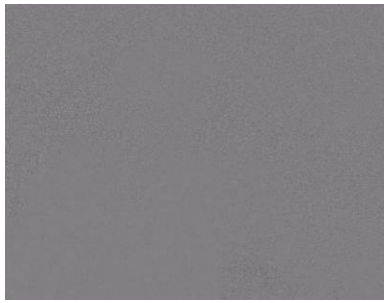


Image difference

Background removal

HDR

And more…

A broad and growing portfolio

The first wave of ST BrightSense devices leveraging ST technologies has arrived

Sensitivity & compactness	**VD55G0** 0.38 MP | 644 x 604 1/9'' format (2.3 mm) Monochrome MIPI CSI-2	**VD56G3** 1.5 MP | 1124 x 1364 1/4'' format (4.6 mm) Monochrome MIPI CSI-2	**VD66GY** 1.5 MP | 1124 x 1364 1/4'' format (4.6 mm) RGB MIPI CSI-2	**VD16GZ** 1.5 MP | 1124 x 1364 1/4'' format (4.6 mm) RGB-IR MIPI CSI-2
Smart & low-power	**VD55G1** 0.56 MP | 804 x 704 1/9'' format (2.3 mm) Monochrome MIPI CSI-2 & I3C			

Leading imaging foundry

Patented technologies

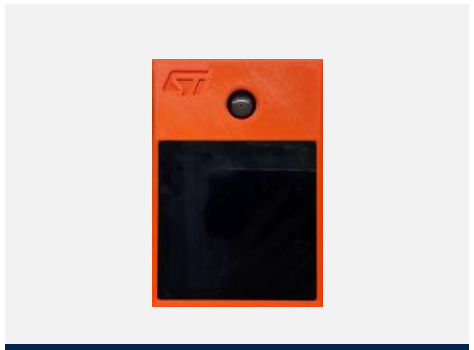
And more soon…

VD55G1
Awarded "Best of Sensors"
Optical & Imaging
Sensors Converge 2024

ST BrightSense & STM32N6

A wide range of use cases

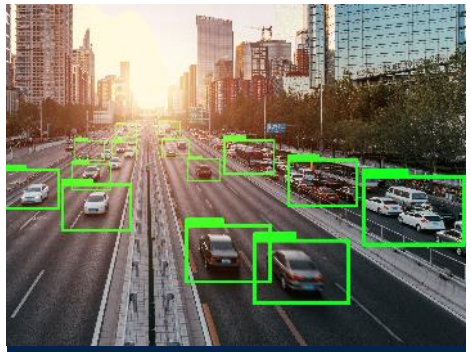
AR/VR/XR



AI assistants

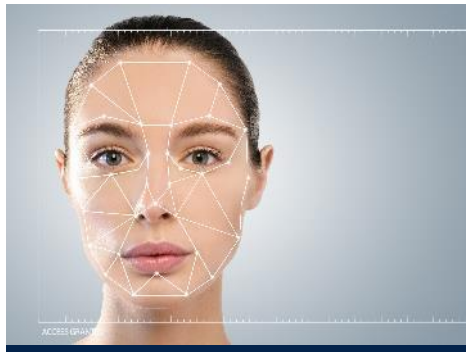
Logistics

Robotics



Transportation

Smart cities



Smart building

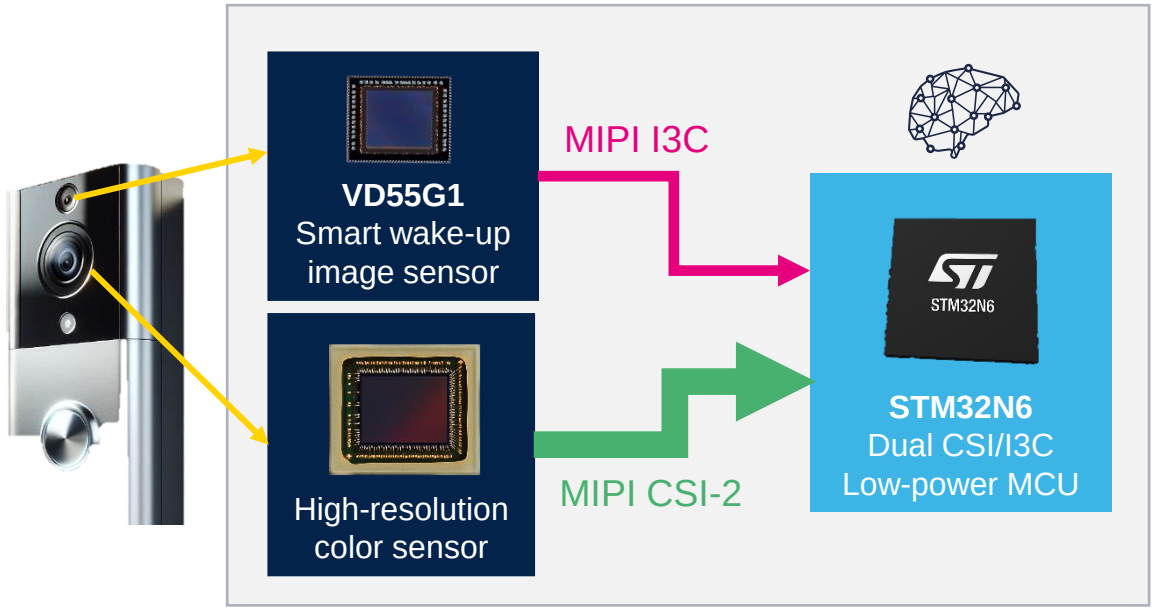
Sports cameras

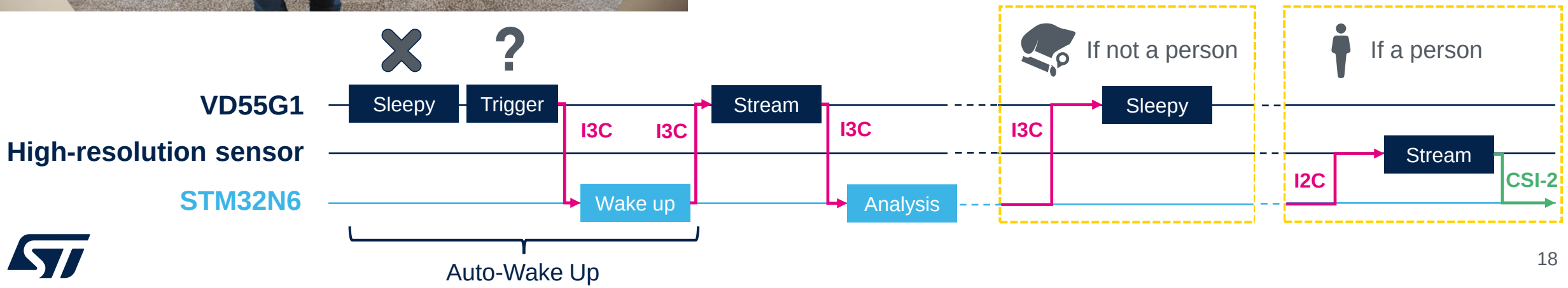
Here are typical use cases, for which ST got requests for combining an ST BrightSense image sensor with STM32N6 to develop smart low-power systems, and for which we plan to develop demonstrations or reference designs.

Enabling smart always-on vision at low-power

With STM32N6 and ST BrightSense

Example of always-on doorbell camera

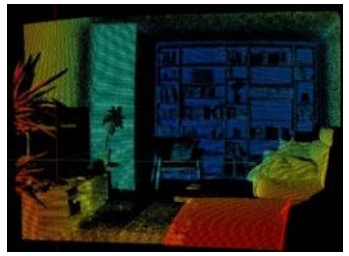




Enabling smart vision with multi-sensing

With STM32N6, ST BrightSense & FlightSense

Example of vacuum cleaning robots



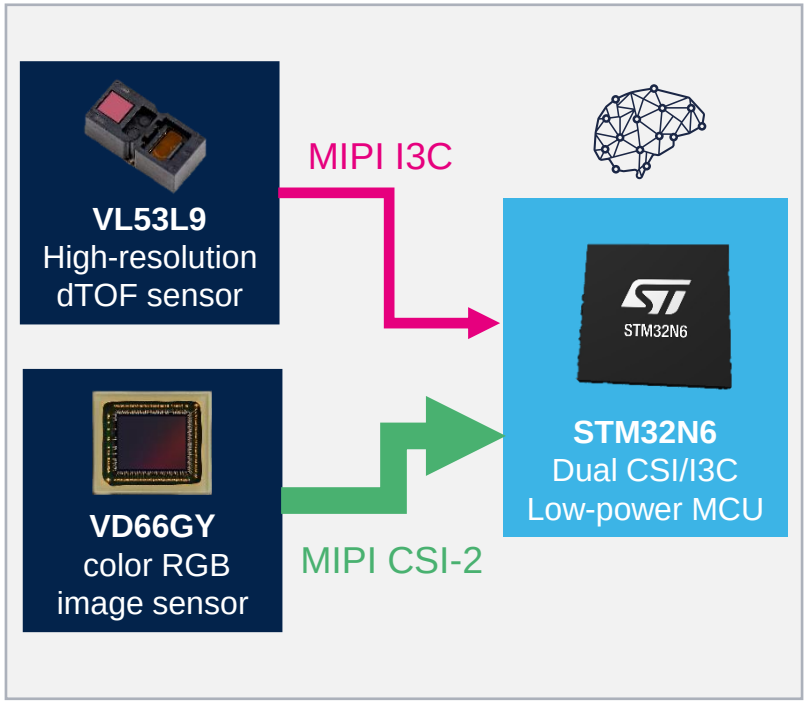
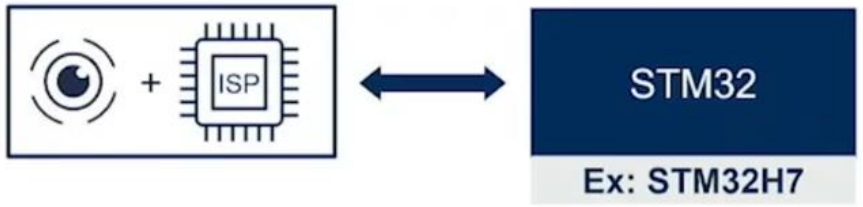


Image Sensor Solutions (Keaton) Cloud Connectivity Control

Image Processor Evolution: Migrating to the Application

Camera module
(from previous generations of MCU)



- The **camera module** embeds
  - The lens
  - The Bayer matrix and filter
  - The Image Signal processor
- Common **output format** of digital cameras:
  - YUV
  - RGB
  - RAW Bayer

Sensor module
(for next generations of MCU)

- The **sensor module** embeds:
  - The lens
  - The Bayer matrix and filter
- The **output format** of sensor modules:
  - RAW Bayer (8, 10, 12, 14 bits)

More versatility to change image sensor
(ISP to tune once)

Totally offload CPU from multi-channel application scenarios
(H264+NPU/NPU+CPU/NPU+DISPLAY)

Cost saving on image sensor

Support **MIPI CSI2 image sensors** (higher bandwidth)

Real-world Use Case

Nice picture for human vision

Usefull picture for computer vision

COMPUTER VISION:

- **Clarify over aesthetics**
- **Relevance over beauty**

Keep in mind

While aesthetics can be important for human perception, computer vision algorithms are primarily concerned with **extracting meaningful information** from images, **regardless of their artistic merit**

AI Examples with STM32N6 and /IOTCONNECT

STM32N6 AI Projects Enabled with /IOTCONNECT

/IOTCONNECT®

Projects on GitHub

n6-ai-getstarted package

- **image classification:** efficientnet_v2B1_240_fft_qdq_int8.onnx
- **instance segmentation:** yolov8n_256_quant_pc_uf_seg_coco-st.tflite
- **object detection:** quantized_tiny_yolo_v2_224_.tflite
- **pose estimation:** st_movenet_lightning_heatmaps_192_int8_pc.tflite

x-cube-n6-ai-multi-pose-estimation-v1.0.0

- yolov8n_256_quant_pc_uf_pose_coco-st_OE_3_1_0.onnx

x-cube-n6-ai-hand-landmarks-v1.0.0

- 033_hand_landmark_full_quant_pc_uf_handl.tflite
- 033_palm_detection_full_quant_pc_uf_od.tflite

x-cube-n6-ai-people-detection-v1.0.0

- quantized_tiny_yolo_v2_224_.tflite

STM32N6_AI_H264_UVC_Application_v0.4.1

- quantized_tiny_yolo_v2_224_.tflite

AI Model Application Example: Crosswalk

AI Model	Core Question	Typical Applications (Crosswalk Scenario)
1. Image Classification	"What general context does this scene show?"	Quick scene labeling (e.g., "busy intersection," "pedestrian crowd").
2. Object Detection	"Which specific objects are present, and where?"	Detection of individual vehicles, pedestrians, and bicycles.
3. People Detection (Subset of Object Detection)	"How many people are here specifically?"	Precise pedestrian counting at intersections; safety monitoring.
4. Instance Segmentation	"Where exactly is each object, and what's its precise shape?"	Detailed tracking and separation of pedestrians; monitoring individual behaviors.
5. Semantic Segmentation	"What area do pedestrians or vehicles occupy?"	Measuring crowd density and pedestrian area to optimize safety and infrastructure.
6. Pose Estimation	"What are individual people doing?"	Detecting unusual postures (falls, dangerous behaviors), aiding in pedestrian safety.
7. Multi-person Pose Estimation	"How are multiple people interacting, and what are their actions?"	Analyzing complex group behaviors (e.g., pedestrian flow, conflicts).

Demo Hardware Setup

Required Hardware:

- STM32N6570-DK Discovery Kit - Purchase | Kit Manual | Device Specifications
- PC with Windows 10/11 (Linux is also supported)
- 2x USB Type-C data cables
- 4x Male to Female header jumpers Buy from Newark
- /IOTCONNECT enabled PMOD such as the DA16200 or DA16600
- A USB-to-Serial converter such as the DSD TECH SH-U09C5 USB to TTL UART Converter

Quick Start Guide:

- Follow the Quick Start Guide on GitHub for step-by-step instructions to:
  - Assemble the STM32N6 hardware.
  - Flash the firmware using ST tools.
  - Configure the device for IoTConnect integration.

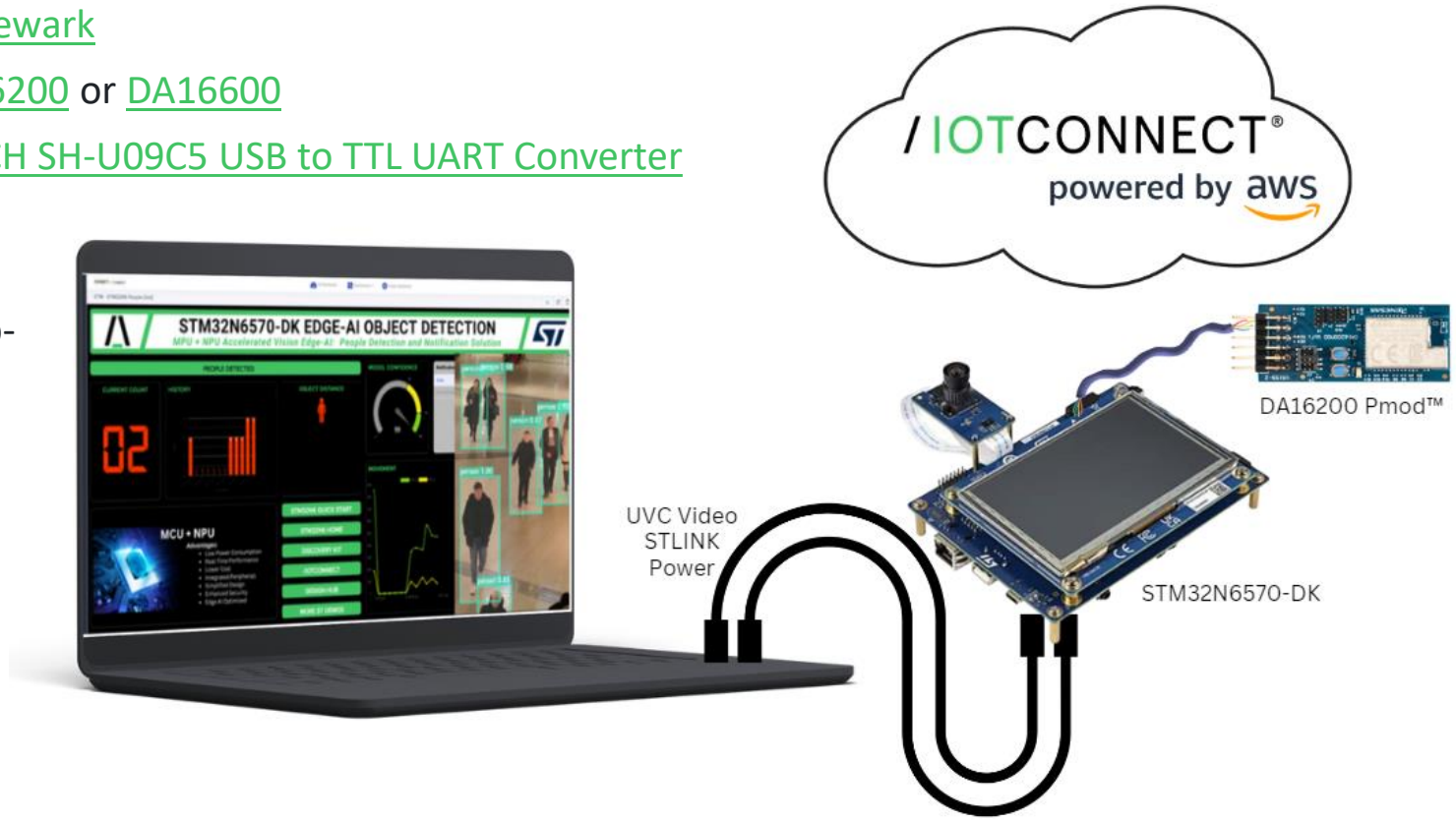


Image Classification

What general context does this scene show?

Image Classification

Model: efficientnet_v2B1_240_fft_qdq_int8.onnx

Key Metrics: 26 FPS | 2.9 MB weights | 1.6 MB activations

What the Model Does

- Analyzes an image and assigns a probability to predefined object categories.

Use Case Scenarios

- **Manufacturing:** Product defect classification
- **Agriculture:** Fruit/vegetable quality sorting
- **Healthcare:** Initial triage of medical images

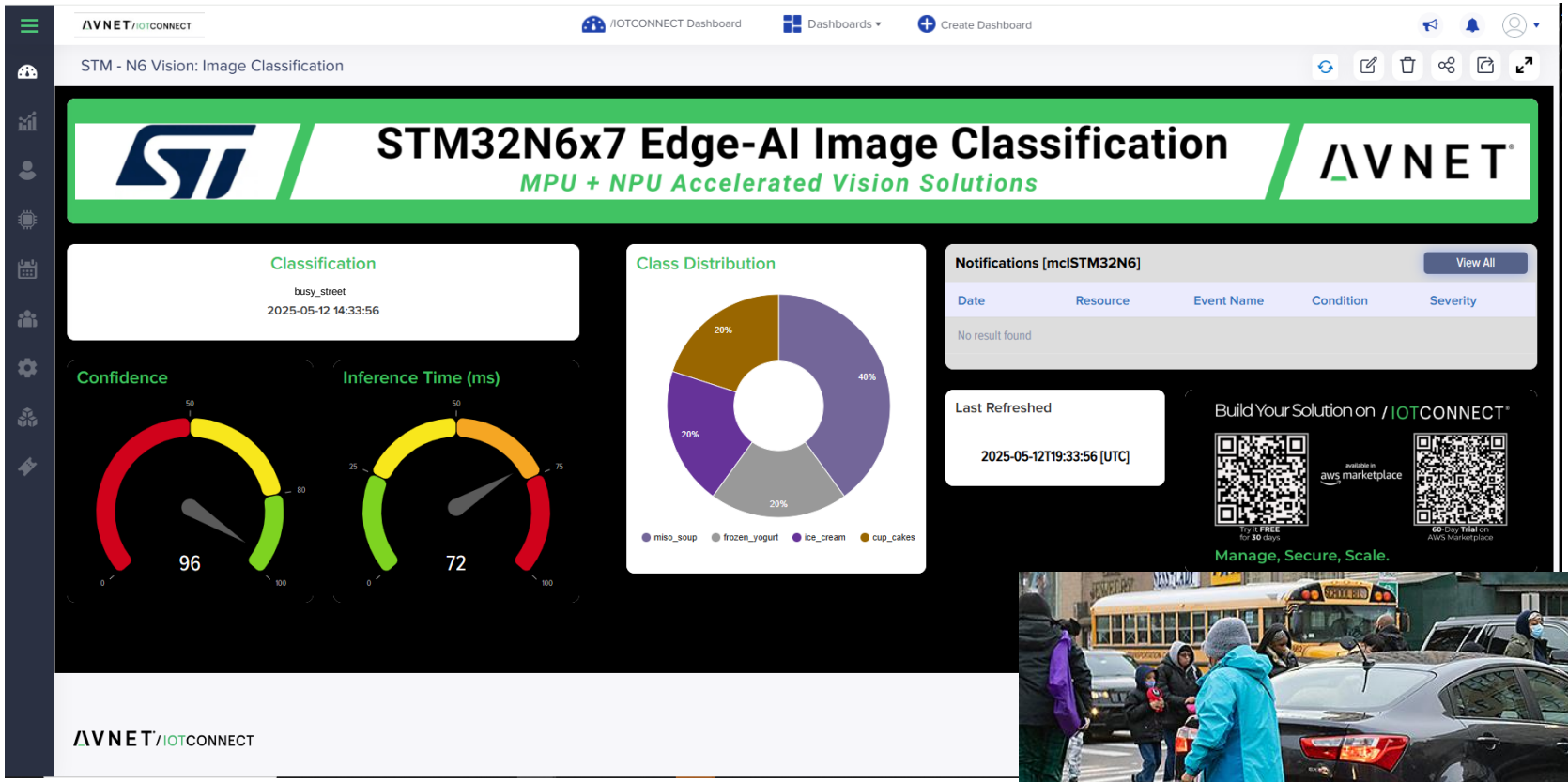
Adaptation

- Domain-specific model retraining
- Real-time predictions (<50 ms)
- Confidence-based reporting

DOG

CAT

Image Classification Example: Crosswalk



/IOTCONNECT Integration

- Sends classification label & confidence
- Monitors performance metrics (inference time)
- Cloud-based telemetry logging for trends

Image Classification: Available Telemetry Attributes

Name	Data Type	Description
inference	decimal	Inference time per frame (ms)
fps	decimal	Frames per second during inference
classification	string	Predicted top class label
confidence	decimal	Confidence score for top class
top_k	int	Number of top classes returned
class1_name	string	Top-1 class label
class1_confidence	decimal	Confidence score for top-1 class
class2_name	string	Top-2 class label
class2_confidence	decimal	Confidence score for top-2 class
class3_name	string	Top-3 class label
class3_confidence	decimal	Confidence score for top-3 class
class4_name	string	Top-4 class label
class4_confidence	decimal	Confidence score for top-4 class
class5_name	string	Top-5 class label
class5_confidence	decimal	Confidence score for top-5 class
		⋮
classN_name	string	Class label for class N
classN_confidence	decimal	Confidence score for class N

Object Detection

Which specific objects are present, and where?

Object / People Detection – Real-Time Situational Awareness

Model: quantized_tiny_yolo_v2_224_.tflite

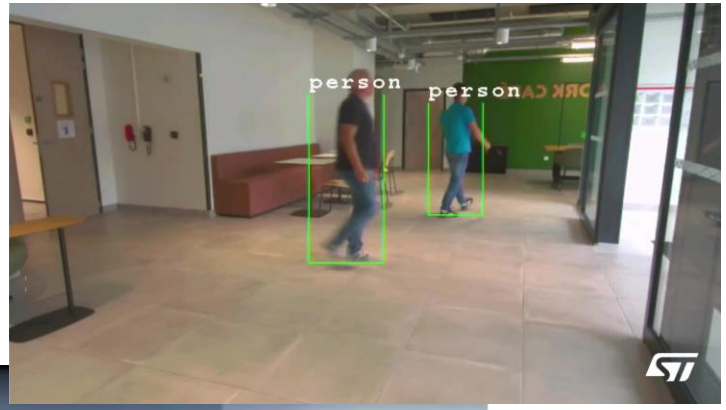
Key Metrics: 26 FPS | 2.9 MB weights | 1.6 MB activations

What the Model Does

- Detects multiple objects in an image by identifying bounding boxes and classifying each with a label and confidence score.

Use Case Scenarios

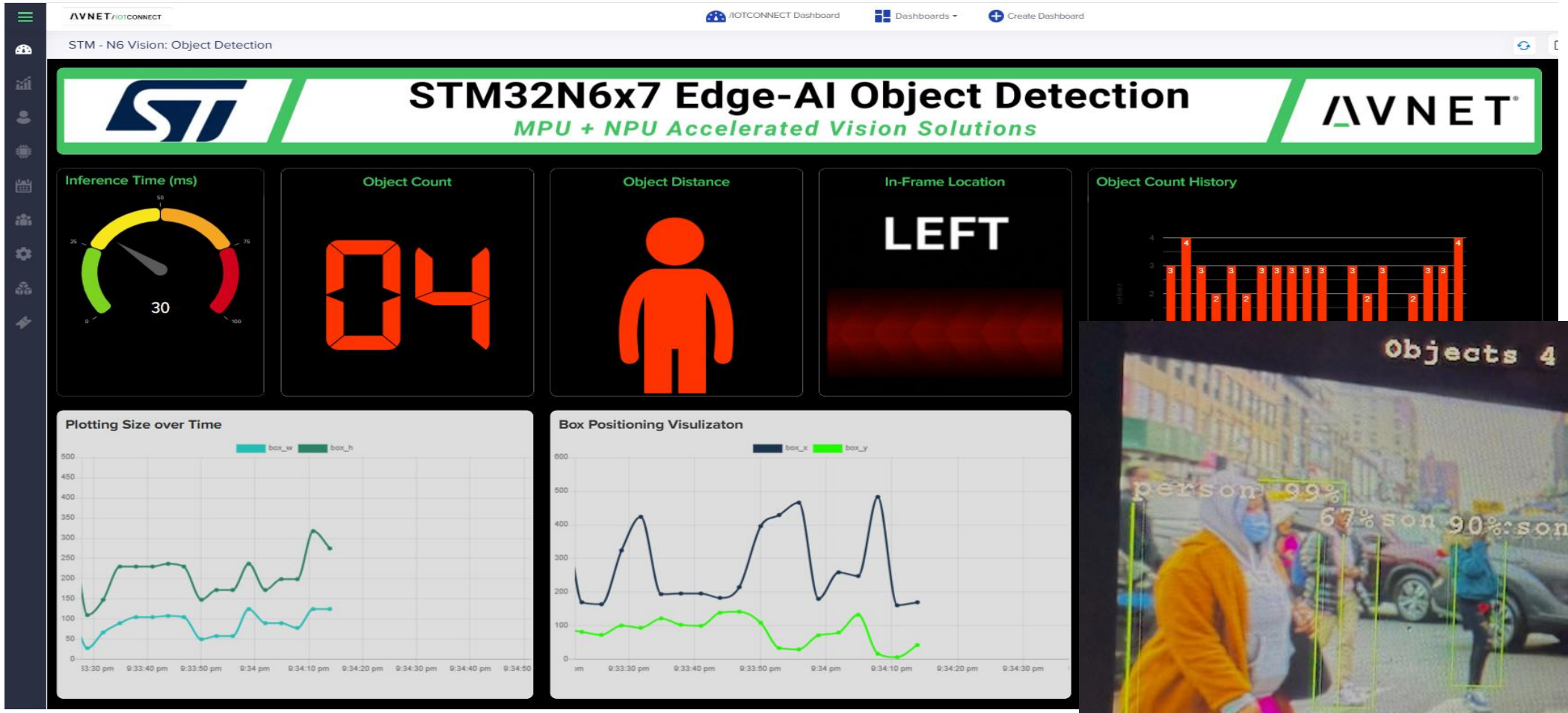
- **Traffic Monitoring:** Detect cars, pedestrians, and bikes for smart city analytics
- **Asset Tracking:** Identify and monitor equipment or inventory in warehouses
- **Wildlife Monitoring:** Detect and track animals in conservation or farming zones
- **Smart Buildings:** Optimize HVAC and lighting
- **Retail:** Track foot traffic and customer flow
- **Security**: Intrusion detection in restricted areas



Adaptation

- Compact and fast model for embedded edge use
- Confidence and bounding box coordinates transmitted, not raw images
- Flexible model retraining for customer-specific object sets

Object Detection Example



/IOTCONNECT Features

- Transmits object class, box coordinates, and confidence
- View object counts and positions remotely
- Detect anomalies or trigger events from the cloud

Object Detection: Available Telemetry Attributes

Name	Data Type	Description
inference	decimal	Inference time per frame (ms)
fps	decimal	Frames per second during inference
object_count	int	Number of detected objects in the frame
dominant_class	string	Most frequent object class in the frame
object1_id	int	Unique object ID 1
object1_class	string	Class label for object 1
object1_bbox_x	decimal	Bounding box X for object 1
object1_bbox_y	decimal	Bounding box Y for object 1
object1_bbox_w	decimal	Bounding box width for object 1
object1_bbox_h	decimal	Bounding box height for object 1
object1_conf	decimal	Confidence score for object 1
object2_id	int	Unique object ID 2
object2_class	string	Class label for object 2
object2_bbox_x	decimal	Bounding box X for object 2
object2_bbox_y	decimal	Bounding box Y for object 2
object2_bbox_w	decimal	Bounding box width for object 2
object2_bbox_h	decimal	Bounding box height for object 2
object2_conf	decimal	Confidence score for object 2
object3_id	int	Unique object ID 3
object3_class	string	Class label for object 3
object3_bbox_x	decimal	Bounding box X for object 3
object3_bbox_y	decimal	Bounding box Y for object 3
object3_bbox_w	decimal	Bounding box width for object 3
object3_bbox_h	decimal	Bounding box height for object 3
object3_conf	decimal	Confidence score for object 3

Name	Data Type	Description
object4_id	int	Unique object ID 4
object4_class	string	Class label for object 4
object4_bbox_x	decimal	Bounding box X for object 4
object4_bbox_y	decimal	Bounding box Y for object 4
object4_bbox_w	decimal	Bounding box width for object 4
object4_bbox_h	decimal	Bounding box height for object 4
object4_conf	decimal	Confidence score for object 4
object5_id	int	Unique object ID 5
object5_class	string	Class label for object 5
object5_bbox_x	decimal	Bounding box X for object 5
object5_bbox_y	decimal	Bounding box Y for object 5
object5_bbox_w	decimal	Bounding box width for object 5
object5_bbox_h	decimal	Bounding box height for object 5
object5_conf	decimal	Confidence score for object 5
⋮		
objectN_id	int	Unique object ID N
objectN_class	string	Class label for object N
objectN_bbox_x	decimal	Bounding box X for object N
objectN_bbox_y	decimal	Bounding box Y for object N
objectN_bbox_w	decimal	Bounding box width for object N
objectN_bbox_h	decimal	Bounding box height for object N
objectN_conf	decimal	Confidence score for object N

Instance Segmentation

Where exactly is each object, and what's its precise shape?

Instance Segmentation – Object Isolation

Model: yolov8n_256_quant_pc_uf_seg_coco-st.tflite

Key Metrics: 24–26 FPS | 3.5 MB weights | 2.7 MB activations

What the Model Does

- Identifies and segments each object in a scene at the pixel level, assigning unique masks per instance.

Use Case Scenarios

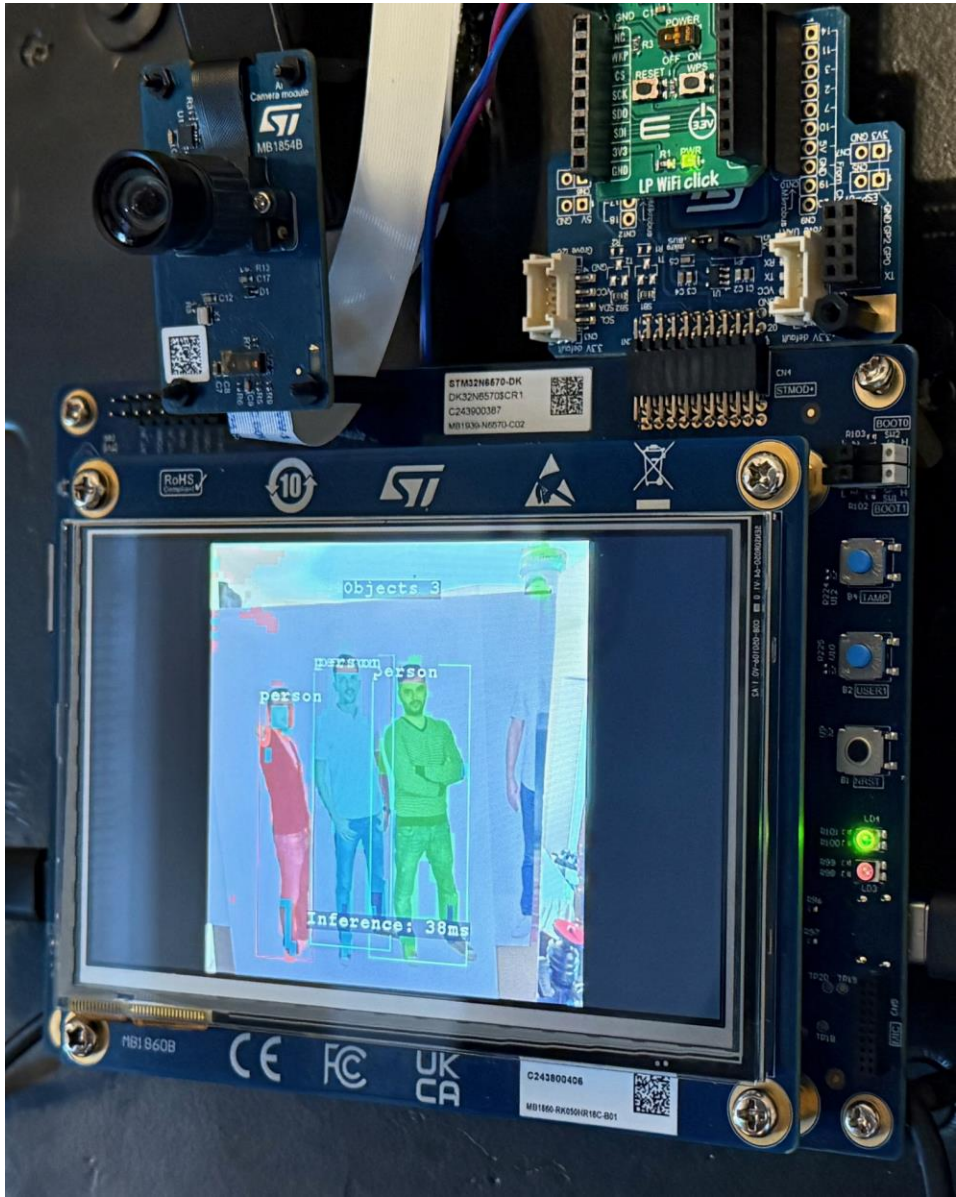
- **Retail Inventory:** Count and separate items on shelves for stock monitoring
- **Autonomous Robotics:** Distinguish individual objects for safe interaction
- **Waste Sorting:** Detect and classify recyclables for automated separation

Adaptation

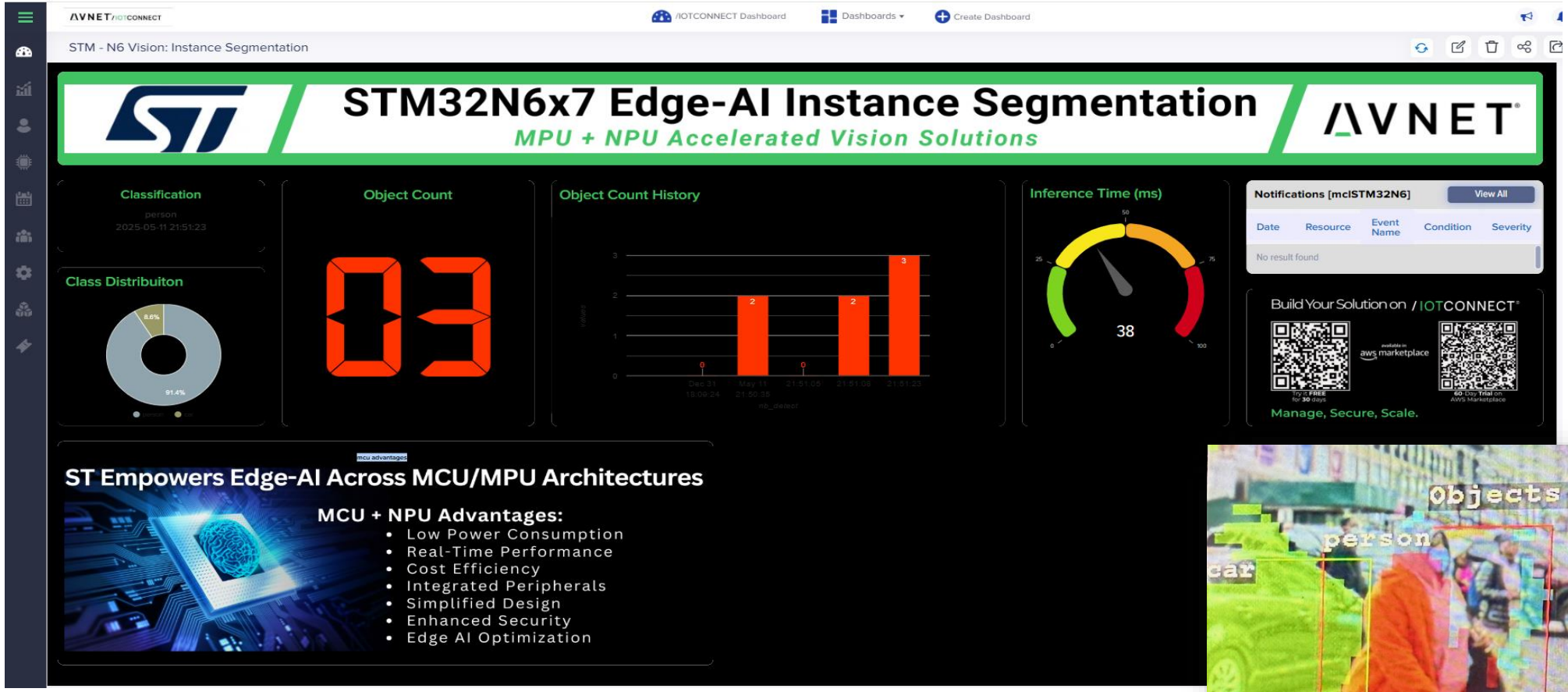
- Generates object masks instead of just bounding boxes
- Enhanced visual understanding for robotics and AR/VR applications
- Model supports consistent class detection before reporting (e.g., “person,” “bottle”)

/IOTCONNECT Integration

- Sends segmentation results and class labels
- Tracks object count and dominant categories
- OTA deployment of updated object masks and segmentation thresholds



Instance Segmentation Example: Crosswalk



/IOTCONNECT Advantage

- Sends detected masks & associated labels
- Ideal for quality inspection or occupancy mapping

Instance Segmentation: Available Telemetry Attributes

Name	Data Type	Description
inference	decimal	Inference time per frame (ms)
fps	decimal	Frames per second during inference
object_count	int	Number of segmented objects in the frame
dominant_class	string	Most frequent object class in the frame
object1_id	int	Unique object instance ID 1
object1_class	string	Class label for object 1
object1_area	int	Pixel area for object 1
object1_bbox_x	decimal	Bounding box X for object 1
object1_bbox_y	decimal	Bounding box Y for object 1
object1_bbox_w	decimal	Bounding box width for object 1
object1_bbox_h	decimal	Bounding box height for object 1
object1_conf	decimal	Confidence for object 1
object2_id	int	Unique object instance ID 2
object2_class	string	Class label for object 2
object2_area	int	Pixel area for object 2
object2_bbox_x	decimal	Bounding box X for object 2
object2_bbox_y	decimal	Bounding box Y for object 2
object2_bbox_w	decimal	Bounding box width for object 2
object2_bbox_h	decimal	Bounding box height for object 2
object2_conf	decimal	Confidence for object 2
object3_id	int	Unique object instance ID 3
object3_class	string	Class label for object 3
object3_area	int	Pixel area for object 3
object3_bbox_x	decimal	Bounding box X for object 3
object3_bbox_y	decimal	Bounding box Y for object 3
object3_bbox_w	decimal	Bounding box width for object 3
object3_bbox_h	decimal	Bounding box height for object 3
object3_conf	decimal	Confidence for object 3

Name	Data Type	Description
object4_id	int	Unique object instance ID 4
object4_class	string	Class label for object 4
object4_area	int	Pixel area for object 4
object4_bbox_x	decimal	Bounding box X for object 4
object4_bbox_y	decimal	Bounding box Y for object 4
object4_bbox_w	decimal	Bounding box width for object 4
object4_bbox_h	decimal	Bounding box height for object 4
object4_conf	decimal	Confidence for object 4
object5_id	int	Unique object instance ID 5
object5_class	string	Class label for object 5
object5_area	int	Pixel area for object 5
object5_bbox_x	decimal	Bounding box X for object 5
object5_bbox_y	decimal	Bounding box Y for object 5
object5_bbox_w	decimal	Bounding box width for object 5
object5_bbox_h	decimal	Bounding box height for object 5
object5_conf	decimal	Confidence for object 5
⋮		
objectN_id	int	Unique object instance ID N
objectN_class	string	Class label for object N
objectN_area	int	Pixel area for object N
objectN_bbox_x	decimal	Bounding box X for object N
objectN_bbox_y	decimal	Bounding box Y for object N
objectN_bbox_w	decimal	Bounding box width for object N
objectN_bbox_h	decimal	Bounding box height for object N
objectN_conf	decimal	Confidence for object N

Semantic Segmentation

What area does an object class occupy?

Semantic Segmentation

Model: deeplab_v3_mobilenetv2_05_16_320_fft_qdq_int8.onnx

Key Metrics: 45 ms inference time | ~3.6 MB weights | ~2.0 MB activations

What the Model Does

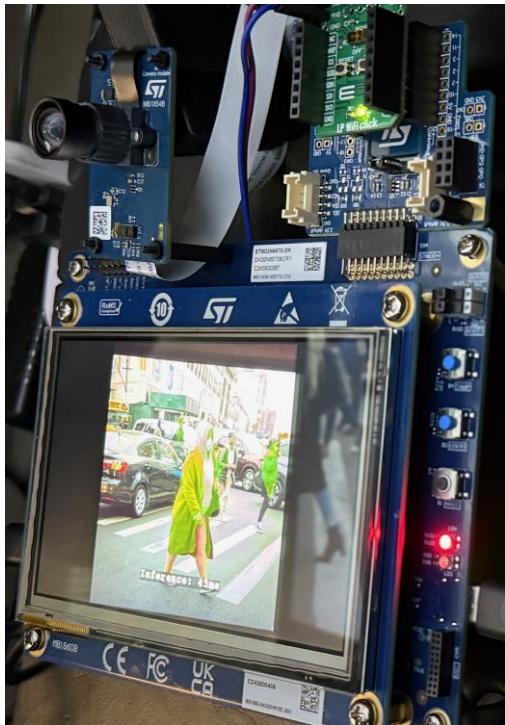
- Classifies each pixel in an image into a semantic category (e.g., road, person, floor), generating a full segmentation map. It can also calculate the total highlighted area in pixels based on a selected class (e.g., person coverage)

Use Case Scenarios

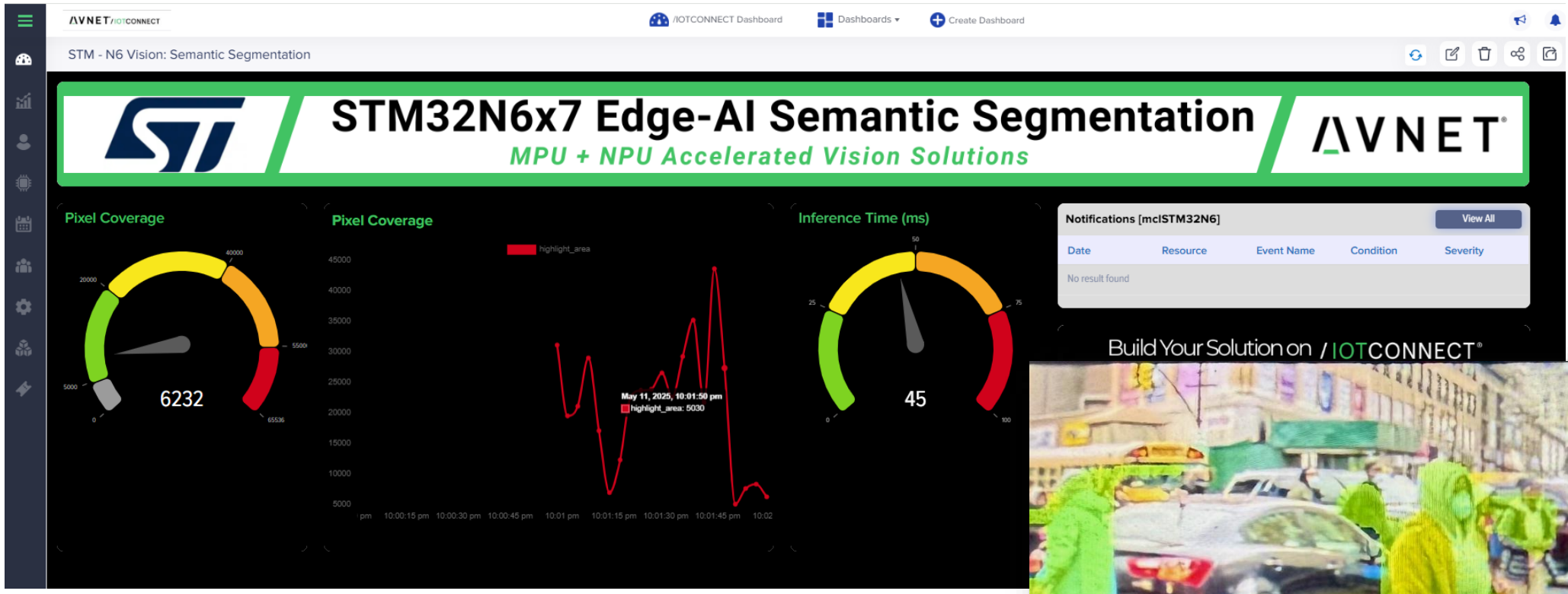
- **Autonomous Navigation:** Scene parsing for obstacle avoidance
- **Smart Cleaning Robots:** Identify floor vs. walls vs. furniture
- **Construction & Retail:** Track space usage, human presence, or hazard zones

Adaptation

- Works well even with small input resolutions (e.g., 320×320)
- Model optimized for fast inference on embedded NPUs
- "Highlight Area" feature quantifies how much space a category (e.g., people) occupies



Semantic Segmentation Example: Crosswalk



Cloud Telemetry via /IOTCONNECT

- Sends highlight_area for monitored category
- Sends inference time in ms for system performance tracking
- Enables threshold-based alerts (exceeds limit = overcrowding)

Semantic Segmentation: Available Telemetry Attributes

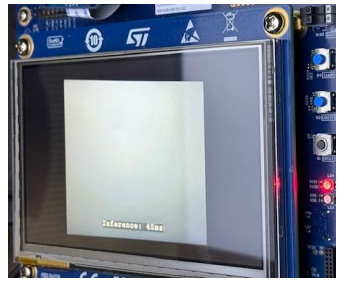
Name	Data Type	Description
inference	decimal	Inference time per frame (ms)
fps	decimal	Frames per second during inference
highlight_area	int	Pixel count for primary (target) class
dominant_class	string	Most frequent class in the frame
class1_id	int	Class index for class 1
class1_name	string	Label name for class 1
class1_area	int	Number of pixels for class 1
class1_percent	decimal	Percent of frame for class 1
class2_id	int	Class index for class 2
class2_name	string	Label name for class 2
class2_area	int	Number of pixels for class 2
class2_percent	decimal	Percent of frame for class 2
class3_id	int	Class index for class 3
class3_name	string	Label name for class 3
class3_area	int	Number of pixels for class 3
class3_percent	decimal	Percent of frame for class 3
class4_id	int	Class index for class 4
class4_name	string	Label name for class 4
class4_area	int	Number of pixels for class 4
class4_percent	decimal	Percent of frame for class 4
⋮		
classN_id	int	Class index for class N
classN_name	string	Label name for class N
classN_area	int	Number of pixels for class N
classN_percent	decimal	Percent of frame for class N

Name	Data Type	Description
object1_id	int	Unique object instance ID 1
object1_class	string	Class label for object 1
object1_area	int	Pixel area for object 1
object1_bbox_x	decimal	Bounding box X for object 1
object1_bbox_y	decimal	Bounding box Y for object 1
object1_bbox_w	decimal	Bounding box width for object 1
object1_bbox_h	decimal	Bounding box height for object 1
object1_conf	decimal	Confidence for object 1
object2_id	int	Unique object instance ID 2
object2_class	string	Class label for object 2
object2_area	int	Pixel area for object 2
object2_bbox_x	decimal	Bounding box X for object 2
object2_bbox_y	decimal	Bounding box Y for object 2
object2_bbox_w	decimal	Bounding box width for object 2
object2_bbox_h	decimal	Bounding box height for object 2
object2_conf	decimal	Confidence for object 2
object3_id	int	Unique object instance ID 3
object3_class	string	Class label for object 3
object3_area	int	Pixel area for object 3
object3_bbox_x	decimal	Bounding box X for object 3
object3_bbox_y	decimal	Bounding box Y for object 3
object3_bbox_w	decimal	Bounding box width for object 3
object3_bbox_h	decimal	Bounding box height for object 3
object3_conf	decimal	Confidence for object 3
⋮		
objectN_id	int	Unique object instance ID N
objectN_class	string	Class label for object N
objectN_area	int	Pixel area for object N
objectN_bbox_x	decimal	Bounding box X for object N
objectN_bbox_y	decimal	Bounding box Y for object N
objectN_bbox_w	decimal	Bounding box width for object N
objectN_bbox_h	decimal	Bounding box height for object N
objectN_conf	decimal	Confidence for object N

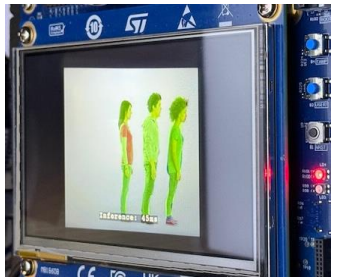
Semantic Segmentation Application Example:

Queue Counter

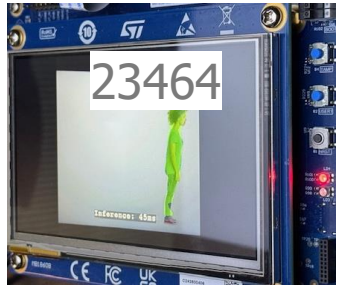
Semantic Segmentation Use Case: Queue Length



53 pixels / 45ns



32,372 pixels / 44ns



10,474 pixels / 45ns

42,972 pixels / 45ns

23,464 pixels / 45ns

60,442 pixels / 45ns

Pose Estimation

What are individual people doing?

Pose Estimation – Real-Time Human Posture Tracking

Model: st_movenet_lightning_heatmaps_192_int8_pc.tflite

Key Metrics: 26 FPS | 3.2 MB weights | 1.8 MB activations

What the Model Does

- Detects and maps 17 human body keypoints (e.g., head, shoulders, elbows, knees) from live video frames using heatmap-based inference.

Use Case Scenarios

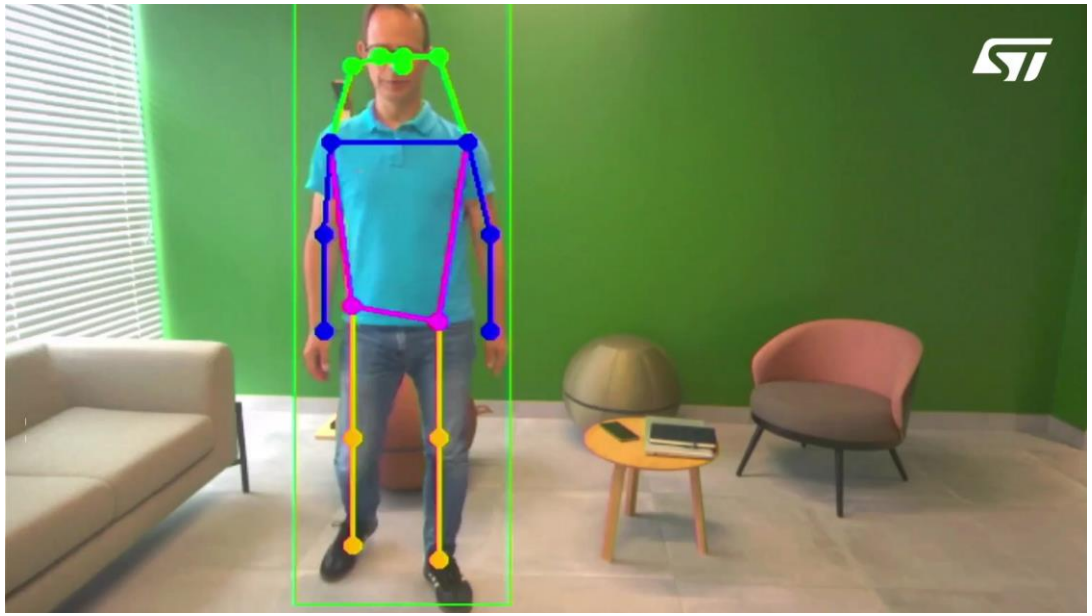
- **Physical Therapy:** Guided rehabilitation with posture feedback
- **Sports Training:** Real-time motion tracking and form analysis
- **Workplace Ergonomics:** Prevent injury by enforcing safe movement patterns

Adaptation

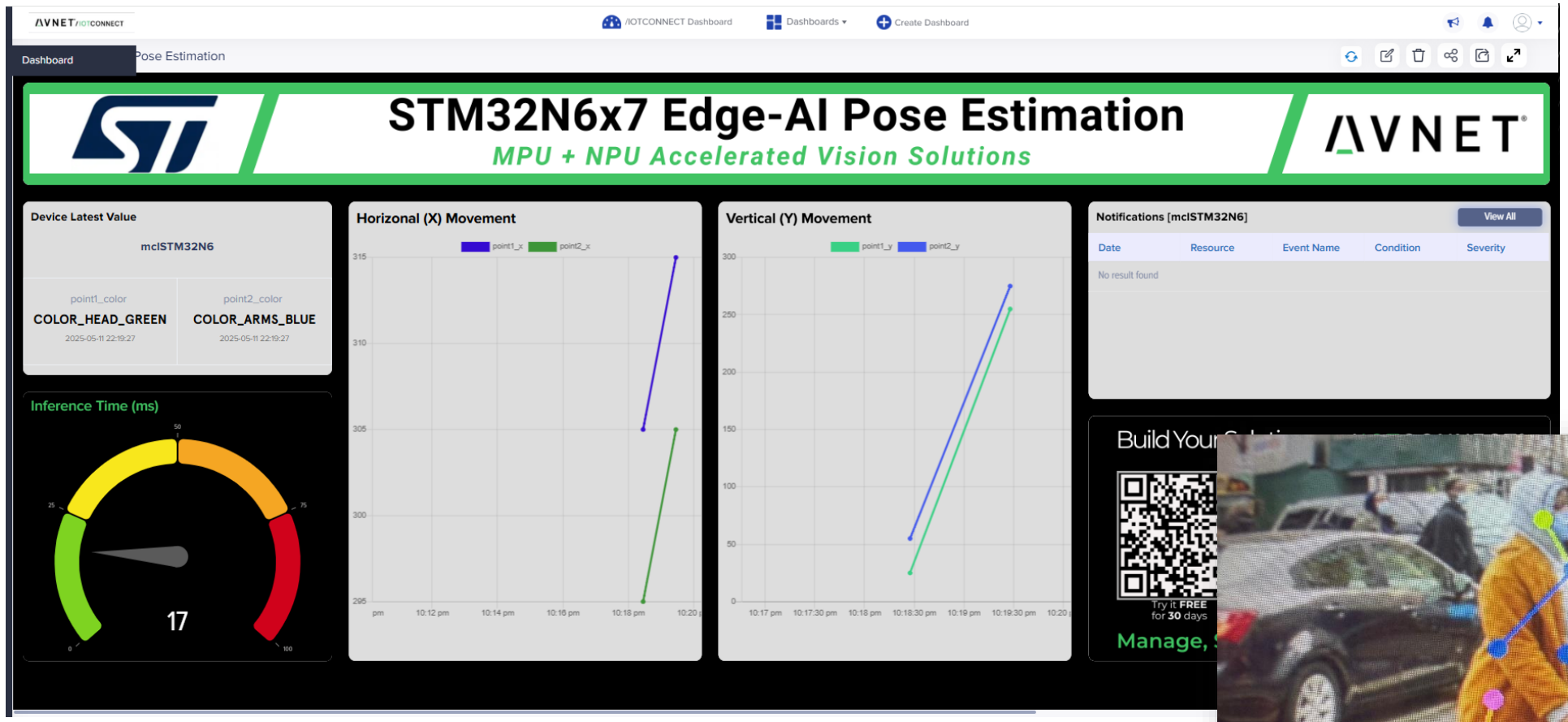
- Heatmap outputs enable smooth, accurate tracking of fast movements
- Easily combined with action classification for event-based alerts
- Suitable for single-person tracking in constrained environments

/IOTCONNECT Integration

- Logs joint positions and posture states
- Triggers alerts for risky or non-compliant movements
- OTA deployment of posture classification thresholds or tracking algorithms



Pose Estimation Example: Crosswalk



/IOTCONNECT Features

- Reports key point coordinates to cloud
- Enables visualization of posture/movement
- Tracks detection confidence over time

Pose Estimation: Available Telemetry Attributes

Name	Data Type	Description
inference	decimal	Inference time per frame (ms)
fps	decimal	Frames per second during inference
person_count	int	Number of people detected in the frame
nose_x	decimal	X coordinate of nose
nose_y	decimal	Y coordinate of nose
nose_conf	decimal	Confidence of nose detection
left_eye_x	decimal	X coordinate of left_eye
left_eye_y	decimal	Y coordinate of left_eye
left_eye_conf	decimal	Confidence of left_eye detection
right_eye_x	decimal	X coordinate of right_eye
right_eye_y	decimal	Y coordinate of right_eye
right_eye_conf	decimal	Confidence of right_eye detection
left_ear_x	decimal	X coordinate of left_ear
left_ear_y	decimal	Y coordinate of left_ear
left_ear_conf	decimal	Confidence of left_ear detection
right_ear_x	decimal	X coordinate of right_ear
right_ear_y	decimal	Y coordinate of right_ear
right_ear_conf	decimal	Confidence of right_ear detection
left_shoulder_x	decimal	X coordinate of left_shoulder
left_shoulder_y	decimal	Y coordinate of left_shoulder

Name	Data Type	Description
left_hip_x	decimal	X coordinate of left_hip
left_hip_y	decimal	Y coordinate of left_hip
left_hip_conf	decimal	Confidence of left_hip detection
right_hip_x	decimal	X coordinate of right_hip
right_hip_y	decimal	Y coordinate of right_hip
right_hip_conf	decimal	Confidence of right_hip detection
left_knee_x	decimal	X coordinate of left_knee
left_knee_y	decimal	Y coordinate of left_knee
left_knee_conf	decimal	Confidence of left_knee detection
right_knee_x	decimal	X coordinate of right_knee
right_knee_y	decimal	Y coordinate of right_knee
right_knee_conf	decimal	Confidence of right_knee detection
left_ankle_x	decimal	X coordinate of left_ankle
left_ankle_y	decimal	Y coordinate of left_ankle
left_ankle_conf	decimal	Confidence of left_ankle detection
right_ankle_x	decimal	X coordinate of right_ankle
right_ankle_y	decimal	Y coordinate of right_ankle
right_ankle_conf	decimal	Confidence of right_ankle detection
left_hip_x	decimal	X coordinate of left_hip
left_hip_y	decimal	Y coordinate of left_hip
left_hip_conf	decimal	Confidence of left_hip detection
right_hip_x	decimal	X coordinate of right_hip
right_hip_y	decimal	Y coordinate of right_hip
right_hip_conf	decimal	Confidence of right_hip detection

Multi-Person Pose Estimation

How are multiple people interacting, and what are their actions?

Multi-Person Pose Estimation – Behavior & Safety

Model: yolov8n_256_quant_pc_uf_pose_coco-st.onnx

Key Metrics: 26 FPS | 3.35 MB weights | 2.59 MB activations

What the Model Does

- Identifies and tracks human body keypoints such as head, shoulders, arms, and legs..

Use Case Scenarios

- **Fitness Equipment:** Posture feedback for training
- **Manufacturing:** Ergonomic compliance tracking
- **Healthcare:** Elderly monitoring and fall detection

Adaptation

- Pose heatmap overlays
- Complex action recognition
- Posture violation alerts

/IOTCONNECT Integration

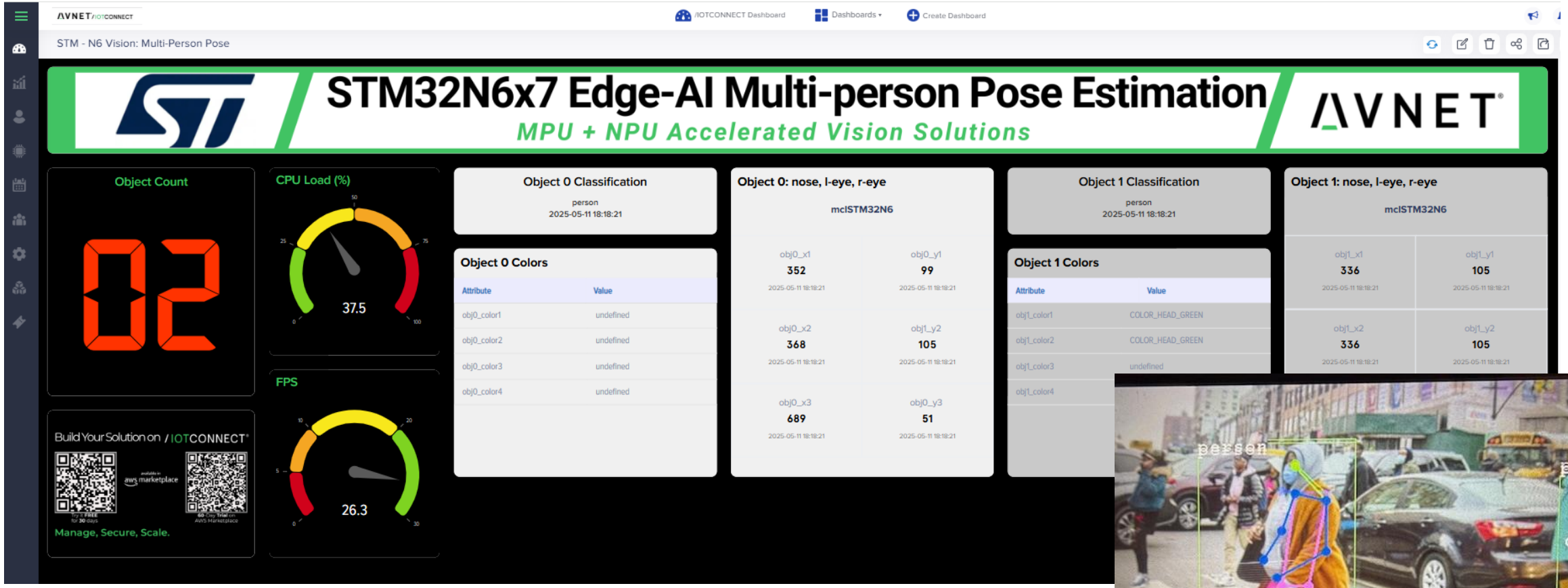
- Logs body position data
- Visualization dashboards
- Safety rule enforcement and alerts

Data Interpretation

yolov8_pose detects **17** standard keypoints per detected person.
The order and meaning of these points follow the standard COCO keypoint convention

- Head (Green)
  - 0: Nose
  - 1: Left Eye
  - 2: Right Eye
  - 3: Left Ear
  - 4: Right Ear
- Arms (Blue)
  - 5: Left Shoulder
  - 6: Right Shoulder
  - 7: Left Elbow
  - 8: Right Elbow
  - 9: Left Wrist
  - 10: Right Wrist
- Torso (Magenta)
  - 11: Left Hip
  - 12: Right Hip
- Legs (Yellow)
  - 13: Left Knee
  - 14: Right Knee
  - 15: Left Ankle
  - 16: Right Ankle

Multi-Person Pose Estimation Example: Crosswalk



/IOTCONNECT Features

- Reports keypoint coordinates to cloud
- Enables visualization of posture/movement
- Tracks detection confidence over time

Multi-Pose Estimation: Available Telemetry Attributes

Name	Data Type	Description
inference	decimal	Inference time per frame (ms)
fps	decimal	Frames per second during inference
person_count	int	Number of people detected in the frame
person1_nose_x	decimal	X coordinate of nose for person 1
person1_nose_y	decimal	Y coordinate of nose for person 1
person1_nose_conf	decimal	Confidence of nose for person 1
person1_left_eye_x	decimal	X coordinate of left_eye for person 1
person1_left_eye_y	decimal	Y coordinate of left_eye for person 1
person1_left_eye_conf	decimal	Confidence of left_eye for person 1
person1_right_eye_x	decimal	X coordinate of right_eye for person 1
person1_right_eye_y	decimal	Y coordinate of right_eye for person 1
person1_right_eye_conf	decimal	Confidence of right_eye for person 1
person1_left_ear_x	decimal	X coordinate of left_ear for person 1
person1_left_ear_y	decimal	Y coordinate of left_ear for person 1
person1_left_ear_conf	decimal	Confidence of left_ear for person 1
person1_right_ear_x	decimal	X coordinate of right_ear for person 1
person1_right_ear_y	decimal	Y coordinate of right_ear for person 1
person1_right_ear_conf	decimal	Confidence of right_ear for person 1
person1_left_shoulder_x	decimal	X coordinate of left_shoulder for person 1
person1_left_shoulder_y	decimal	Y coordinate of left_shoulder for person 1
person1_left_shoulder_conf	decimal	Confidence of left_shoulder for person 1
person1_right_shoulder_x	decimal	X coordinate of right_shoulder for person 1
person1_right_shoulder_y	decimal	Y coordinate of right_shoulder for person 1
person1_right_shoulder_conf	decimal	Confidence of right_shoulder for person 1
person1_left_elbow_x	decimal	X coordinate of left_elbow for person 1
person1_left_elbow_y	decimal	Y coordinate of left_elbow for person 1
person1_left_elbow_conf	decimal	Confidence of left_elbow for person 1
person1_right_elbow_x	decimal	X coordinate of right_elbow for person 1
person1_right_elbow_y	decimal	Y coordinate of right_elbow for person 1
person1_right_elbow_conf	decimal	Confidence of right_elbow for person 1
person1_left_wrist_x	decimal	X coordinate of left_wrist for person 1
person1_left_wrist_y	decimal	Y coordinate of left_wrist for person 1
person1_left_wrist_conf	decimal	Confidence of left_wrist for person 1
person1_right_wrist_x	decimal	X coordinate of right_wrist for person 1
person1_right_wrist_y	decimal	Y coordinate of right_wrist for person 1
person1_right_wrist_conf	decimal	Confidence of right_wrist for person 1

Name	Data Type	Description
person1_left_hip_x	decimal	X coordinate of left_hip for person 1
person1_left_hip_y	decimal	Y coordinate of left_hip for person 1
person1_left_hip_conf	decimal	Confidence of left_hip for person 1
person1_right_hip_x	decimal	X coordinate of right_hip for person 1
person1_right_hip_y	decimal	Y coordinate of right_hip for person 1
person1_right_hip_conf	decimal	Confidence of right_hip for person 1
person1_left_knee_x	decimal	X coordinate of left_knee for person 1
person1_left_knee_y	decimal	Y coordinate of left_knee for person 1
person1_left_knee_conf	decimal	Confidence of left_knee for person 1
person1_right_knee_x	decimal	X coordinate of right_knee for person 1
person1_right_knee_y	decimal	Y coordinate of right_knee for person 1
person1_right_knee_conf	decimal	Confidence of right_knee for person 1
person1_left_ankle_x	decimal	X coordinate of left_ankle for person 1
person1_left_ankle_y	decimal	Y coordinate of left_ankle for person 1
person1_left_ankle_conf	decimal	Confidence of left_ankle for person 1
person1_right_ankle_x	decimal	X coordinate of right_ankle for person 1
person1_right_ankle_y	decimal	Y coordinate of right_ankle for person 1
person1_right_ankle_conf	decimal	Confidence of right_ankle for person 1
person2_nose_x	decimal	X coordinate of nose for person 2
person2_nose_y	decimal	Y coordinate of nose for person 2
person2_nose_conf	decimal	Confidence of nose for person 2
person2_left_eye_x	decimal	X coordinate of left_eye for person 2
person2_left_eye_y	decimal	Y coordinate of left_eye for person 2
person2_left_eye_conf	decimal	Confidence of left_eye for person 2
person2_right_eye_x	decimal	X coordinate of right_eye for person 2
person2_right_eye_y	decimal	Y coordinate of right_eye for person 2
person2_right_eye_conf	decimal	Confidence of right_eye for person 2
person2_left_ear_x	decimal	X coordinate of left_ear for person 2
person2_left_ear_y	decimal	Y coordinate of left_ear for person 2
person2_left_ear_conf	decimal	Confidence of left_ear for person 2
person2_right_ear_x	decimal	X coordinate of right_ear for person 2
person2_right_ear_y	decimal	Y coordinate of right_ear for person 2
person2_right_ear_conf	decimal	Confidence of right_ear for person 2
person2_left_shoulder_x	decimal	X coordinate of left_shoulder for person 2
person2_left_shoulder_y	decimal	Y coordinate of left_shoulder for person 2
person2_left_shoulder_conf	decimal	Confidence of left_shoulder for person 2

Name	Data Type	Description
person2_right_shoulder_x	decimal	X coordinate of right_shoulder for person 2
person2_right_shoulder_y	decimal	Y coordinate of right_shoulder for person 2
person2_right_shoulder_conf	decimal	Confidence of right_shoulder for person 2
person2_left_elbow_x	decimal	X coordinate of left_elbow for person 2
person2_left_elbow_y	decimal	Y coordinate of left_elbow for person 2
person2_left_elbow_conf	decimal	Confidence of left_elbow for person 2
person2_right_elbow_x	decimal	X coordinate of right_elbow for person 2
person2_right_elbow_y	decimal	Y coordinate of right_elbow for person 2
person2_right_elbow_conf	decimal	Confidence of right_elbow for person 2
person2_left_wrist_x	decimal	X coordinate of left_wrist for person 2
person2_left_wrist_y	decimal	Y coordinate of left_wrist for person 2
person2_left_wrist_conf	decimal	Confidence of left_wrist for person 2
person2_right_wrist_x	decimal	X coordinate of right_wrist for person 2
person2_right_wrist_y	decimal	Y coordinate of right_wrist for person 2
person2_right_wrist_conf	decimal	Confidence of right_wrist for person 2
person2_left_hip_x	decimal	X coordinate of left_hip for person 2
person2_left_hip_y	decimal	Y coordinate of left_hip for person 2
person2_left_hip_conf	decimal	Confidence of left_hip for person 2
person2_right_hip_x	decimal	X coordinate of right_hip for person 2
person2_right_hip_y	decimal	Y coordinate of right_hip for person 2
person2_right_hip_conf	decimal	Confidence of right_hip for person 2
person2_left_knee_x	decimal	X coordinate of left_knee for person 2
person2_left_knee_y	decimal	Y coordinate of left_knee for person 2
person2_left_knee_conf	decimal	Confidence of left_knee for person 2
person2_right_knee_x	decimal	X coordinate of right_knee for person 2
person2_right_knee_y	decimal	Y coordinate of right_knee for person 2
person2_right_knee_conf	decimal	Confidence of right_knee for person 2
person2_left_ankle_x	decimal	X coordinate of left_ankle for person 2
person2_left_ankle_y	decimal	Y coordinate of left_ankle for person 2
person2_left_ankle_conf	decimal	Confidence of left_ankle for person 2
person2_right_ankle_x	decimal	X coordinate of right_ankle for person 2
person2_right_ankle_y	decimal	Y coordinate of right_ankle for person 2
person2_right_ankle_conf	decimal	Confidence of right_ankle for person 2
⋮		
personN_keypoint_x	decimal	X coordinate for keypoint of person N
personN_keypoint_y	decimal	Y coordinate for keypoint of person N
personN_keypoint_conf	decimal	Confidence for keypoint of person N

Hand Landmark

Tracking a persons' hand and movements in real-time.

Hand Landmark Detection – Touchless Control

Models

- 033_hand_landmark_full_quant_pc_uf_handl.tflite
- 033_palm_detection_full_quant_pc_uf_od.tflite

Key Metric

- Palm Detection: 192×192 | 1.1 MB weights | 1.1 MB activations
- Hand Landmark: 224×224 | 3.2 MB weights | 1.0 MB activations

What the Model Does

- Detects the palm first, then pinpoints 21 hand joint positions (fingers, knuckles, wrist).

Use Case Scenarios

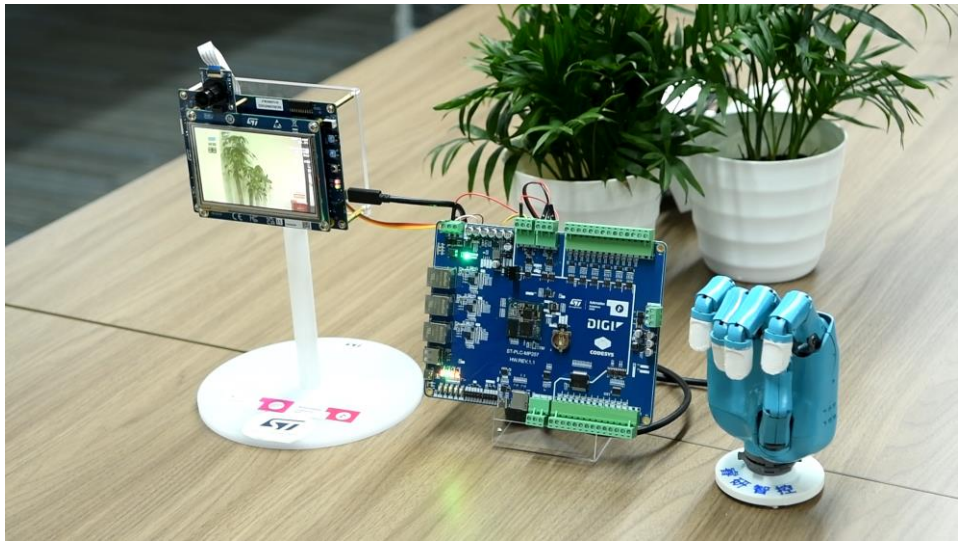
- **Smart Appliances:** Gesture to activate functions
- **Healthcare:** Sterile, non-contact interfaces
- **Public Kiosks:** Hygienic navigation systems

Adaptation

- Custom gesture libraries
- Parallel palm and hand landmark models
- Feedback-triggered interactions

/IOTCONNECT Integration

- Gesture telemetry and event logging
- Remote performance monitoring
- OTA updates for new gesture sets



Hand Landmark Detection: Available IOT Data

Name	Data Type	Description	Range
classification	string	Gesture classification label	N/A
cpu	float	MCU CPU load percentage	0.0- 100.0
nn_fps	float	Neural network inference frames per second	0- 60
nn_period	float	NN execution interval (ms)	~15- 100
inf_time	float	Inference time for palm detection (ms)	0- 50
hl_time	float	Inference time for hand landmark detection (ms)	0- 50
disp_time	float	Display rendering time (ms)	0- 50
hand_rotation	float	Rotation of detected hand (degrees)	-180- 180
nb_detect	int	Number of hands detected	0- 2
conf	float	Detection confidence of primary hand	0.0- 1.0
hand_valid	int	Boolean flag for valid hand detection	0 or 1
hand_x	float	Hand bounding box center X coordinate	0- frame width
hand_y	float	Hand bounding box center Y coordinate	0- frame height
hand_width	float	Width of detected hand bounding box	0- frame width
hand_height	float	Height of detected hand bounding box	0- frame height
wrist_x	float	X coordinate of wrist landmark	0- frame width
wrist_y	float	Y coordinate of wrist landmark	0- frame height
thumb_cmc_x	float	X coordinate of thumb CMC joint	0- frame width
thumb_cmc_y	float	Y coordinate of thumb CMC joint	0- frame height
thumb_mcp_x	float	X coordinate of thumb MCP joint	0- frame width
thumb_mcp_y	float	Y coordinate of thumb MCP joint	0- frame height
thumb_ip_x	float	X coordinate of thumb IP joint	0- frame width
thumb_ip_y	float	Y coordinate of thumb IP joint	0- frame height
thumb_tip_x	float	X coordinate of thumb tip	0- frame width
thumb_tip_y	float	Y coordinate of thumb tip	0- frame height

Name	Data Type	Description	Range
index_mcp_x	float	X coordinate of index MCP joint	0- frame width
index_mcp_y	float	Y coordinate of index MCP joint	0- frame height
index_pip_x	float	X coordinate of index PIP joint	0- frame width
index_pip_y	float	Y coordinate of index PIP joint	0- frame height
index_dip_x	float	X coordinate of index DIP joint	0- frame width
index_dip_y	float	Y coordinate of index DIP joint	0- frame height
index_tip_x	float	X coordinate of index fingertip	0- frame width
index_tip_y	float	Y coordinate of index fingertip	0- frame height
middle_mcp_x	float	X coordinate of middle MCP joint	0- frame width
middle_mcp_y	float	Y coordinate of middle MCP joint	0- frame height
middle_pip_x	float	X coordinate of middle PIP joint	0- frame width
middle_pip_y	float	Y coordinate of middle PIP joint	0- frame height
middle_dip_x	float	X coordinate of middle DIP joint	0- frame width
middle_dip_y	float	Y coordinate of middle DIP joint	0- frame height
middle_tip_x	float	X coordinate of middle fingertip	0- frame width
middle_tip_y	float	Y coordinate of middle fingertip	0- frame height
ring_mcp_x	float	X coordinate of ring MCP joint	0- frame width
ring_mcp_y	float	Y coordinate of ring MCP joint	0- frame height
ring_pip_x	float	X coordinate of ring PIP joint	0- frame width
ring_pip_y	float	Y coordinate of ring PIP joint	0- frame height
ring_dip_x	float	X coordinate of ring DIP joint	0- frame width
ring_dip_y	float	Y coordinate of ring DIP joint	0- frame height
ring_tip_x	float	X coordinate of ring fingertip	0- frame width
ring_tip_y	float	Y coordinate of ring fingertip	0- frame height
pinky_mcp_x	float	X coordinate of pinky MCP joint	0- frame width
pinky_mcp_y	float	Y coordinate of pinky MCP joint	0- frame height
pinky_pip_x	float	X coordinate of pinky PIP joint	0- frame width
pinky_pip_y	float	Y coordinate of pinky PIP joint	0- frame height
pinky_dip_x	float	X coordinate of pinky DIP joint	0- frame width
pinky_dip_y	float	Y coordinate of pinky DIP joint	0- frame height
pinky_tip_x	float	X coordinate of pinky fingertip	0- frame width
pinky_tip_y	float	Y coordinate of pinky fingertip	0- frame height

AI Tools

ST AI Optimization Tools

ST Edge AI model zoo

OR

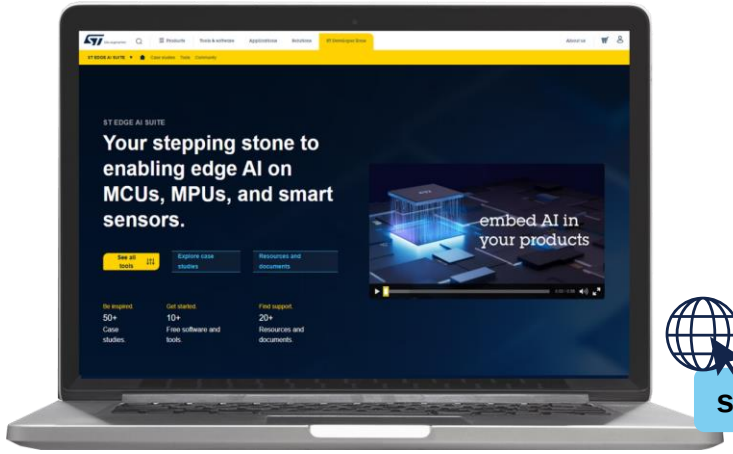
Bring Your Own Model

Keras · TensorFlow · PyTorch · MATLAB · scikit learn · ONNX

STM32Cube.AI

OR

ST Edge AI Developer Cloud



st.com/st-edge-ai-suite

Optimize your NN model

Graph optimizer | Quantizer | Memory optimizer

Powered by ST Edge AI Core technology

STM32Cube ecosystem · Command-line interface

Neural network model converter **for STM32 MCU**

Optimize and benchmark your NN model

Graph optimizer | Quantizer | Memory optimizer

Powered by ST Edge AI Core technology

Online platform · REST API · Benchmarking tool

Neural network model converter and benchmarking **for ST products**

STM32 · Stellar · ISPU

Generate optimized code

.c Optimized model

.a AI runtime

STM32 C-code of the optimized model and AI runtime (.a)

Generate optimized code and project files

Optimized model

MX STM32CubeMX IOC file

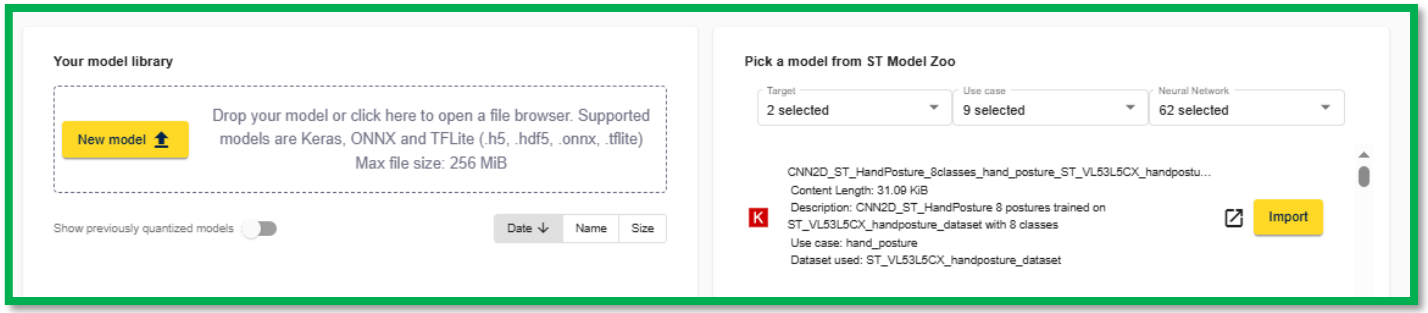
IDE STM32CubeIDE project

Makefile project

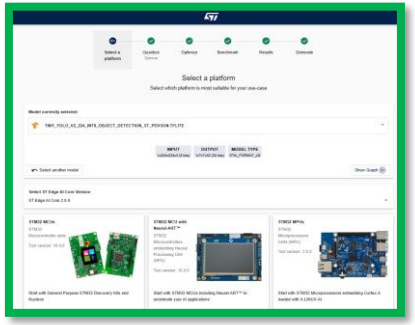
Optimized model code in several formats to suit your needs

ST Edge AI Developer Cloud

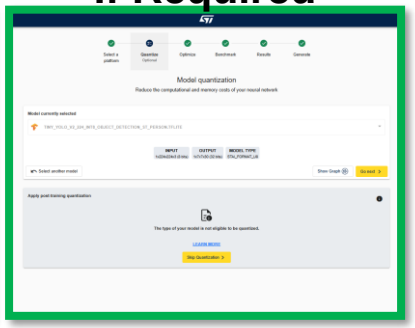
Import or Select AI Model(s)



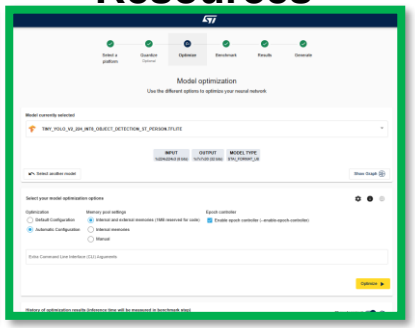
Select HW Platform



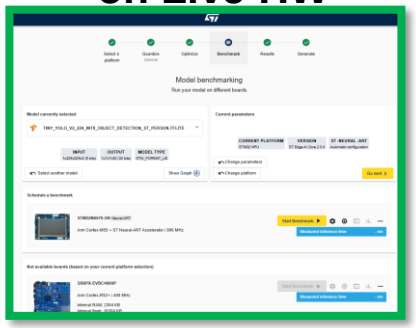
Quantize Model If Required



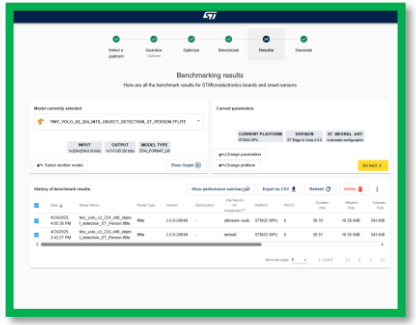
Optimize for Resources



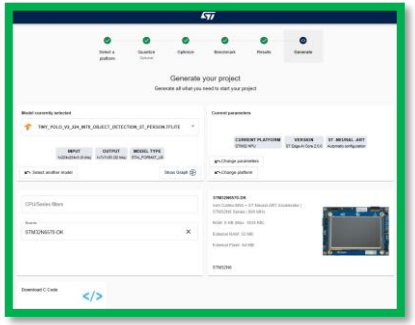
Benchmark on Live HW



Analyze Results



Generate Code



Results

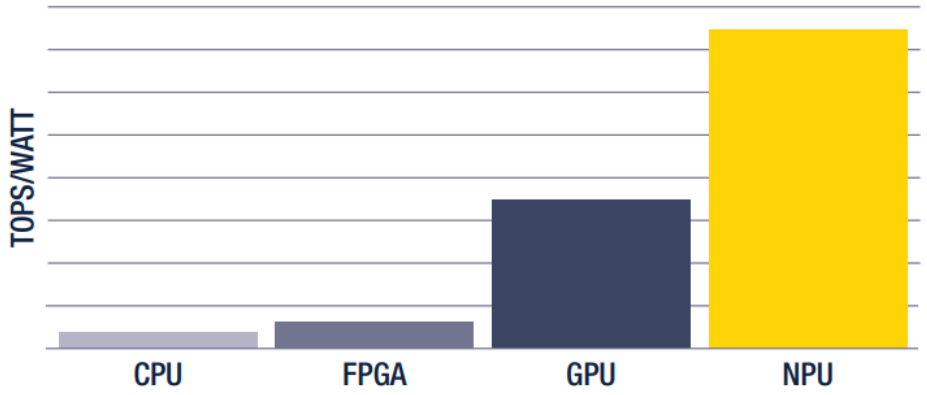
Model Name	Platform	Duration (ms)	Weights Size	Activation Size	Flash size	RAM size	Device	Clock
tiny_yolo_v2_224_int8_object_detection_ST_Person.tflite	STM32 NPU	30.33	10.55 MiB	343 KiB	10.77 MiB	343.01 KiB	STM32N6570-DK	800MHz

ST Neural-ART Accelerator

AI Acceleration Performance by Technology

Feature	CPU	GPU	NPU
Primary function	General-purpose processing	Graphics rendering, parallel computation	Neural network acceleration
Architecture	Few powerful cores, high clock speed	Many smaller cores, SIMD architecture	Specialized cores for neural networks
Processing type	Sequential processing	Parallel processing	Parallel and low-latency processing
Instruction set	Complex instruction set (e.g., x86, Arm)	Graphics and parallel computation instructions	Optimized for convolutional neural network operations
Energy efficiency	Moderate	Moderate to high	High
Use cases	General computing, control tasks	Graphics rendering, scientific simulations, ML training	Edge AI, real-time inference, IoT devices

Embedding NPUs directly into MCUs unlocks unprecedented performance and power efficiency, enabling real-time AI in edge applications previously impossible with traditional embedded processors.



GPU: Graphic accelerator
NPU: Neural processing unit (AI accelerator)

Source W. Dally

Why do you need an NPU?

0.2 FPS

STM32H7

4 FPS

Raspberry PI 4 CPU

30 FPS

STM32N6 with **NPU**

Typical Object Detection NN model: Tiny-YoLoV2 416x416

STM32N6x7 AI-Acceleration Benefit by Model

Model	CPU-bound inferencing on Cortex-M55 @ 400 MHz		Accelerated inferencing on ST Neural-ART Accelerator @ 1 GHz		Improvement with acceleration
	Time (ms)	fps	Time (ms)	fps	
MobileNet v1[1]	2244	0.45	19.4	51.54	x116
MobileNet v2[2]	1385	0.72	21.1	47.45	x66
Tiny Yolo v2[3]	3895	0.26	30.6	32.71	x127
Yolo v8n 256[4]	1821	0.55	31	32.26	x59
Yamnet 1024[5]	252	3.97	9.8	101.7	X26

Note: 1 **Image Classification** - Quantized int8, input resolution 224x224x3, trained on ImageNet dataset. Model footprint: 4.45 MB weights, 1.53 MB activations
2 **Image Classification** - Quantized int8, input resolution 224x224x3, trained on ImageNet dataset. Model footprint: 4.14 MB weights, 2.08 MB activations
3 **Object Detection** - Quantized int8, input resolution 224x224x3, trained on COCO dataset. Model footprint: 10.55 MB weights, 0.38 MB activations
4 **Object Detection** - Quantized int8, input resolution 256x256x3, trained on COCO dataset. Model footprint: 3.05 MB weights, 1.6 MB activations
5 **Audio Event Classification** - Quantized int8, input resolution 64x96, trained on AudioSet dataset. Model footprint: 3.4 MB weights, 0.14 MB activations

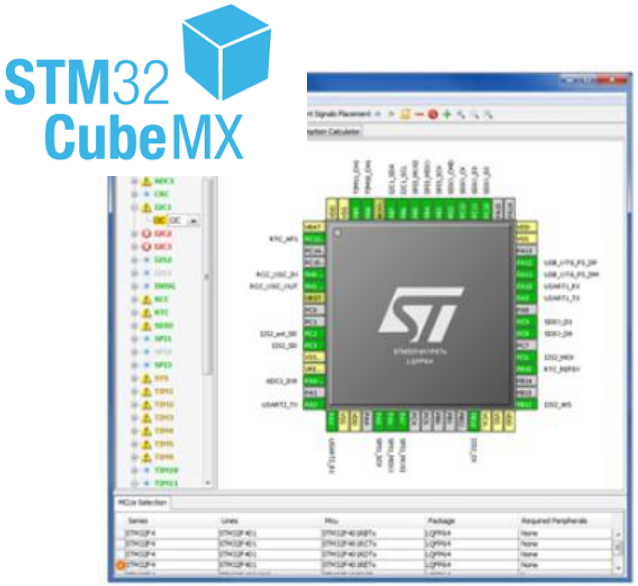
Connectivity

Avnet's X-CUBE-IoTC-DA16K-PMOD

Integration with STMicroelectronics' STM32Cube Ecosystem

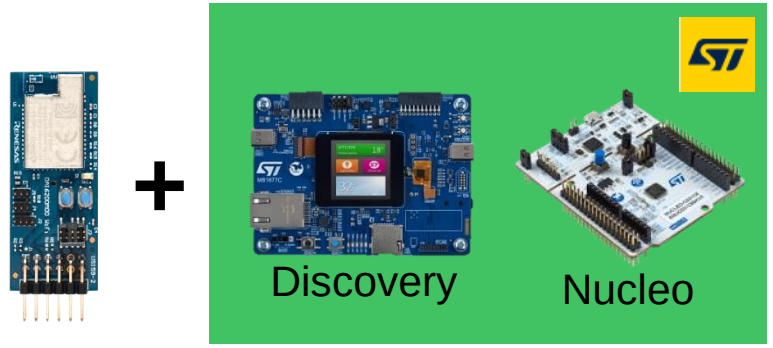
Seamless Integration with STM32Cube Tools

- **STM32CubeMX Support**:
  - Easily configure peripherals, including the **/IOTCONNECT PMOD**, via a graphical interface.
- **STM32CubeIDE Compatibility**:
  - Ready-to-use project templates streamline development within ST's IDE.
- **Middleware and Drivers**:
  - Includes prebuilt libraries to interface with the DA16K PMOD for IoT communication.



Broad STM32 MCU Compatibility

- **Universal Support:**
  - Works with all STM32 MCUs—only requires a **UART interface**.
- **Lightweight Communication**:
  - Utilizes a simple AT command set for ease of integration.



Programming the STM32N6: Signing & Flashing

STM32N6 Project Code Organization

Three Key Binaries:

1. **Bootloader** (ex. ai_fsbl.hex):
   - Initializes system hardware, validates firmware and enforces secure boot.
2. **AI Model** (ex. network_data.hex):
   - Contains optimized model data for Neural-ART accelerator.
3. **Application** (ex. x-cube-n6-ai-h264-usb-uvc.hex):
   - Implements core functionalities (e.g., AI inference, camera processing).

Memory Region	Start Address	End Address	Purpose
External Flash (xSPI)	0x70000000	0x9FFFFFFF	Primary storage for firmware binaries and AI model data.
Bootloader (FSBL)	0x70000000	0x900FFFFF	Initializes system hardware and validates firmware for secure boot.
Application	0x70100000	0x901FFFFF	Main application logic, including peripherals and AI functionality.
AI Model Data	0x70200000	0x9027FFFF	Neural network weights and parameters for the STM32N6 Neural-ART accelerator.
SRAM1	0x30000000	0x301FFFFF	Used for runtime application execution, stack, and heap.
SRAM2	0x30200000	0x303FFFFF	Reserved for neural network activations and temporary buffers.
AXI SRAM	0x24000000	0x2407FFFF	High-speed buffers for critical data.
OTP Memory	0x1FFF7800	0x1FFF7A0F	Secure, non-volatile memory for cryptographic keys used in code signing.
Peripheral Registers	0x40000000	0x5FFFFFFF	Memory-mapped I/O for peripherals such as GPIO, UART, and SPI.

Code Signing for STM32N6

Why Code Signing is Important

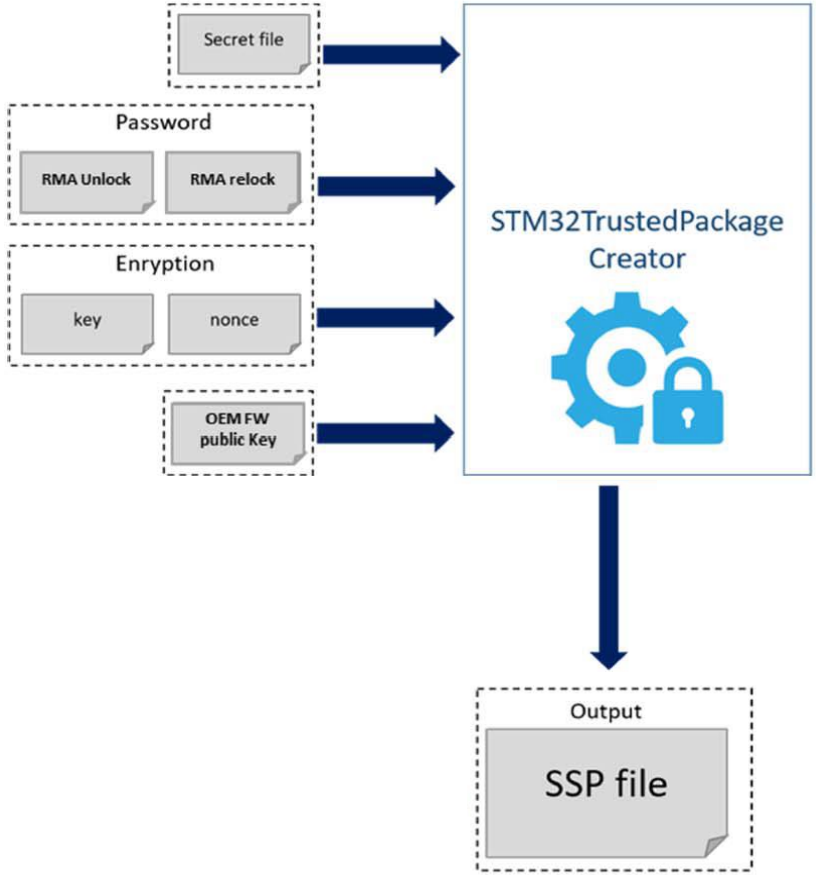
- **Security:** Ensures only authenticated and untampered firmware runs on the device.
- **Trust:** Confirms the firmware originates from a verified source.
- **Compliance:** Meets secure boot requirements for protecting intellectual property and user data.

How Code Signing Works

- **Private Key:** Used to sign the firmware during development.
- **Public Key:** Stored in the device's OTP memory or bootloader for signature verification.
- **Process:**
  - Developer signs the firmware.
  - Bootloader verifies the signature using the public key.
  - Execution only proceeds if the signature is valid.

How It is Done

- Compile the Application
- For Production Builds:
  - Generate the Key / create a private-public RSA key pair
  - Program the OTP
- Run the ST Signing Tool
- Flash the Signed Firmware



User's Manual Link

Flashing the STM32N6 Octo-SPI Flash

Step 1: Compile Code

- Use **STM32CubeIDE** or command-line tools to compile the application
  - Ensure the binary (.bin or .hex) is generated.
  - Verify successful compilation.

Step 2: Sign the Application

- Use the **STM32 Trusted Package Creator** to sign the application.
  - Input: Unsigned application binary.
  - Output: Signed firmware (.sfi).

Step 3: Set Boot Switches

- Configure boot switches for **Development Mode** (BOOT1 = Right).
- Enables the debugger and external loader usage.

Step 4: Connect the Board

- Plug in the STM32N6570-DK via USB Type-C for ST-LINK and power.
  - Ensure both USB connections are stable (one for debug/power, one for programming).

Step 5: Open STM32CubeProgrammer

- Launch the latest version of STM32CubeProgrammer (STM32Prog).
  - Ensure the correct version (e.g., v2.11 or later).

Step 6: Select External Loader

- Go to **External Loader** menu in STM32CubeProgrammer and activate the appropriate loader. (MX66UW1G45G_STM32N6.stldr).

Step 7: Flash the FSBL

- Load the ai_fsbl.hex file into the external flash.
  - Use the **Erasing & Programming** tab.
  - Verify the starting address (e.g., 0x90000000)

Step 8: Flash the AI Models

- Load the network_data.hex file into the external flash. - Program the model data to its specific address (e.g., 0x90200000).

Step 9: Flash the Application

- Load the signed application binary (application.sfi) into the external flash.
  - Program the application at its designated address (e.g., 0x90100000).

Step 10: Reset Boot Switches

- Configure boot switches for **Flash Boot Mode** (BOOT1 = Left).
  - Enables secure boot and execution from external flash.

Step 11: Reboot the Board

- Power cycle the board or press the reset button.
  - Ensure the application executes correctly from flash.

Resources


1800FIX
NO TURNING

Resources

GitHub

- **Cloud-controlled Image Sensor Project**

iotc-isp-tune-stm32/STM32MP257-DK at main · avnet-iotconnect/iotc-isp-tune-stm32

- STM32N6 /IOTCONNECT Quick Start

iotc-stm32-n6-demos/doc/QUICKSTART.md at main · avnet-iotconnect/iotc-stm32-n6-demos