2021 embedded VISION summit®
VIRTUAL | MAY 25-28

DepthAI: Embedded, Performant Spatial AI and CV

Brandon Gilles
Luxonis

LUXonis

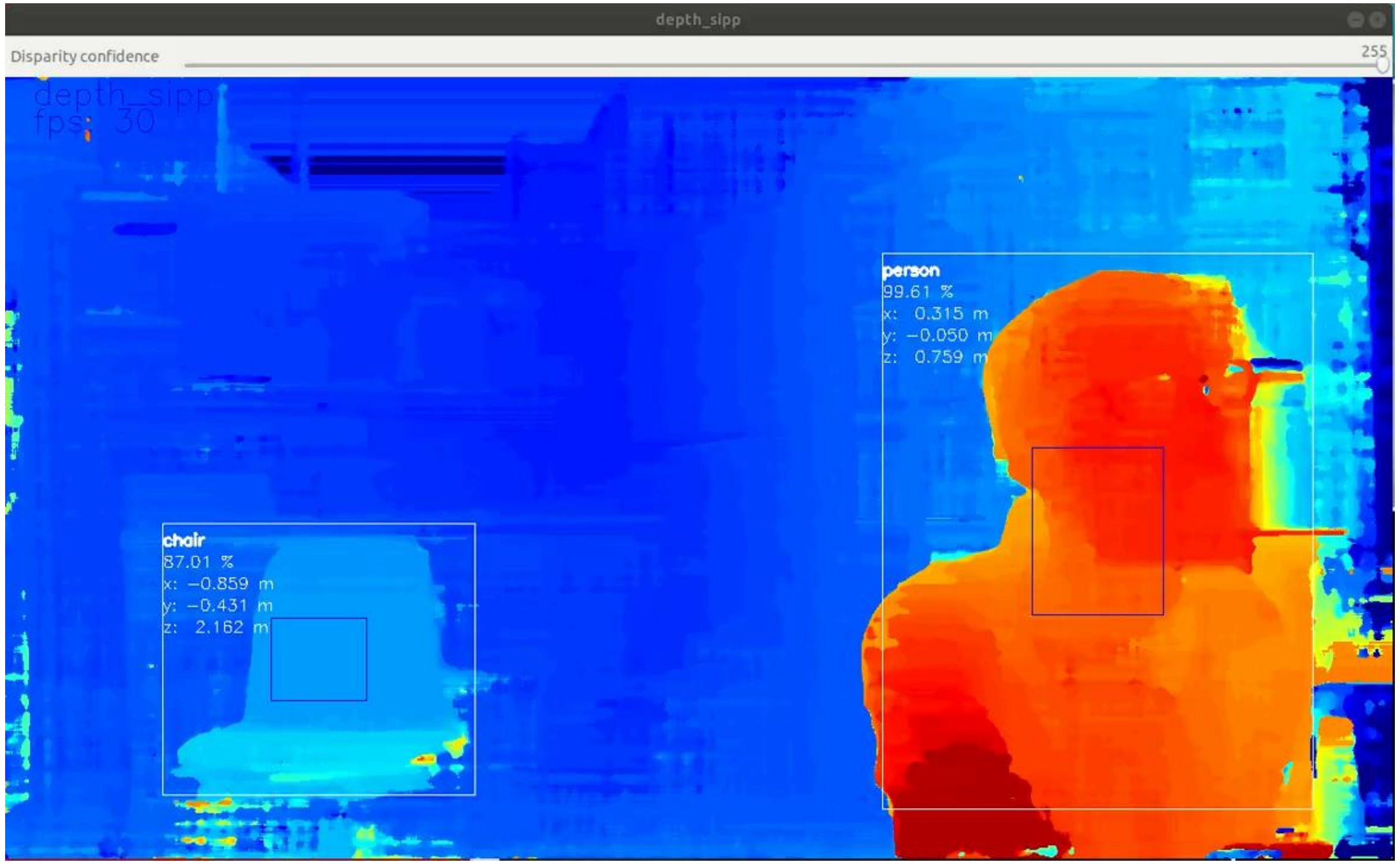
Start with the Why

- Pressing health/safety problem
- Combination 3 available technologies solve the problem:
  - **Spatial sensing** (disparity depth, Lidar)
  - **AI** (neural inference)
  - **CV** (feature extraction tracking, platform pose/motion estimation, etc.)
- But only tractable if on an **embedded system** (low size, weight, power, and cost) and **performant**
- No platform existed that had all 5 of these:
  - Embedded, Performant, Spatial, AI and CV

What is Spatial AI and CV?

- The capability to get neural inference results (e.g. semantic segmentation) in physical space.
    - E.g.: XYZ locations of all the ripe strawberries in real-time.
- Tight fusion of
    - **AI** (object detection, semantic segmentation, etc.)
    - with **CV** (platform motion/pose, object tracking)
    - and **Depth** (disparity)
    - to give **3D position** in world coordinates of detected **objects**, **features**, or **semantic** labels (at the pixel level).

Spatial AI and CV – Basic Example

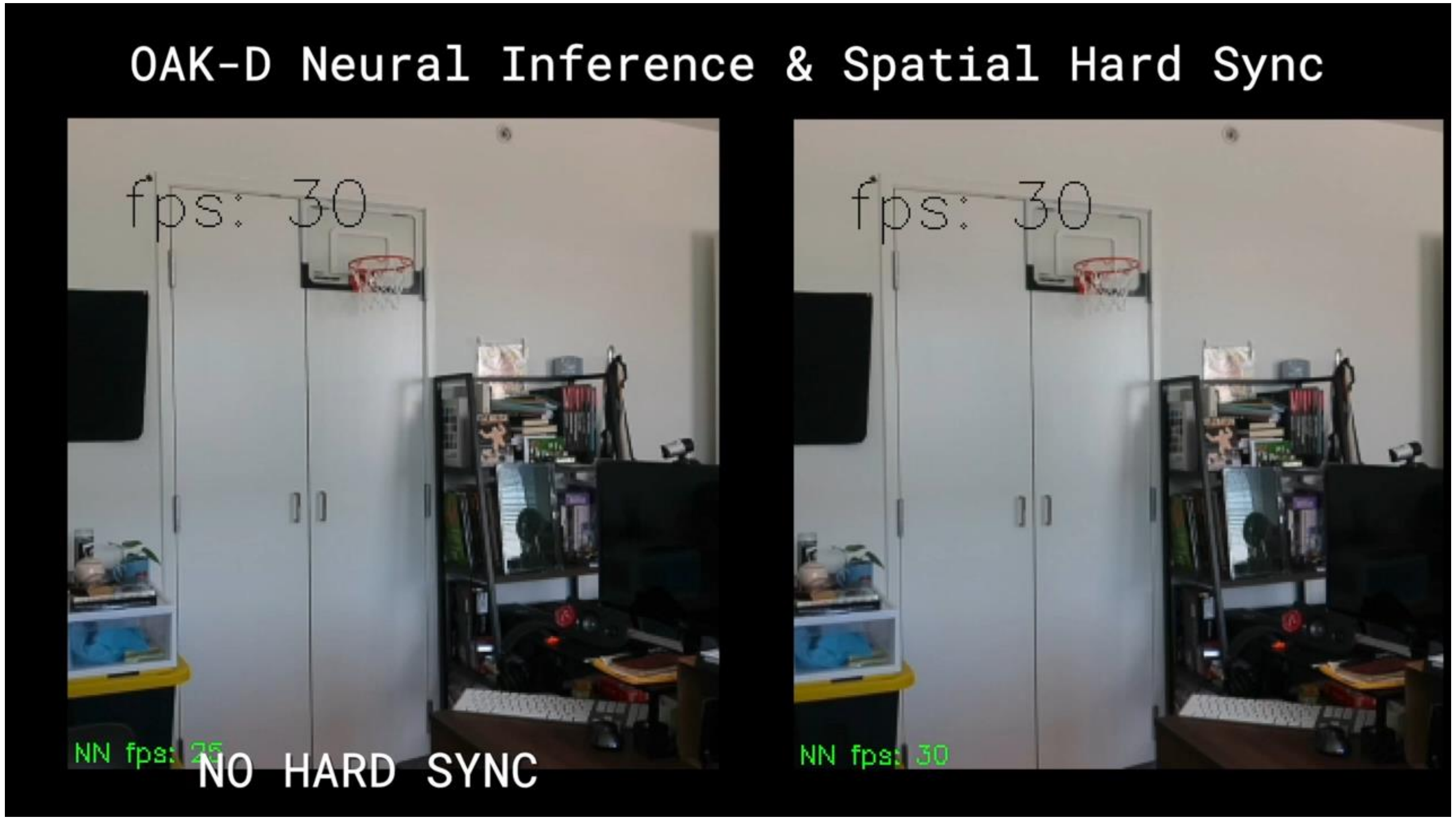


```
person
99.61 %
x:  0.315 m
y: -0.050 m
z:  0.759 m
```

```
chair
87.01 %
x: -0.859 m
y: -0.431 m
z:  2.162 m
```

https://youtu.be/sO1EU5AUq4U

Spatial AI and CV – Fast Motion



https://youtu.be/UQJ6cW7KB34

New Era of Performance on Embedded Systems

- Traditionally, the capability to handle any of the following necessitated a full-fledged operating system-capable computer:
    - High resolution image sensors
    - High frame rates
    - Multiple image sensors
    - Complex AI + CV pipelines
    - Spatial sensing
- This is now all doable on an embedded system
- And this opens up all sorts of applications which were previously intractable

Example: AI-Guided Lossless Zoom

https://youtu.be/H-FjrbWsaKg

Example: AI-Guided Lossless Zoom

https://youtu.be/uylZgG3yLiU

Asymmetric Multi-Processor Embedded System

- Network on Chip (NoC) architecture allows tying together AI/CV/Spatial hardware:
  - 16 Vector processors (think GPU, but architected for computer vision)
  - ~20 fixed-function CV processors (Harris, Canny, warp/dewarp, motion estimation)
  - 2 AI processors
  - 1 semi-global-matching disparity-depth hardware block
- The network on chip is prioritizable, so that it’s selectable which functions drink from the fire-hose of the high-resolution sensor(s)
- Our DepthAI Pipeline Builder allows quick/easy configuration for your application

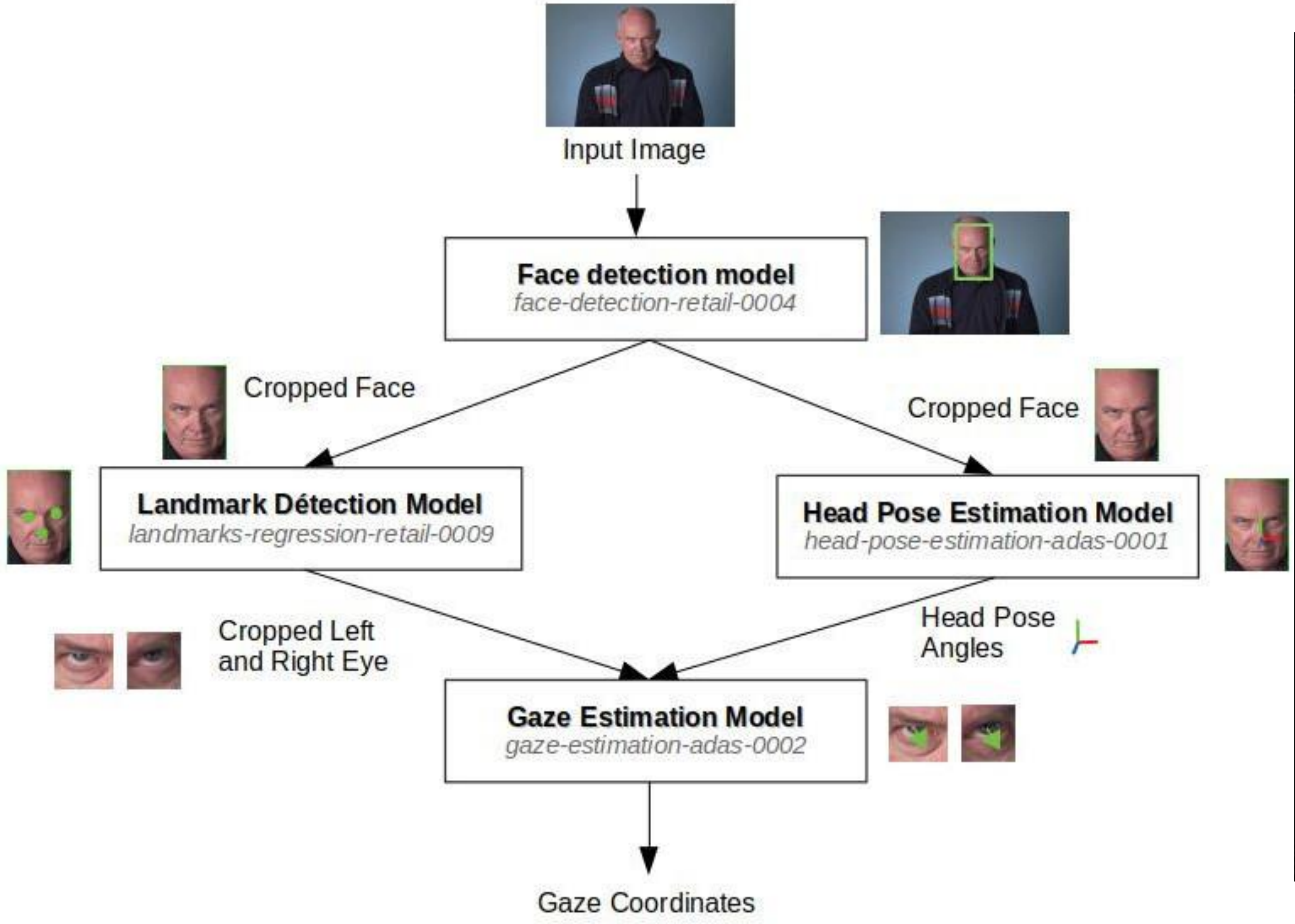
DepthAI Pipeline Builder (Gen2)

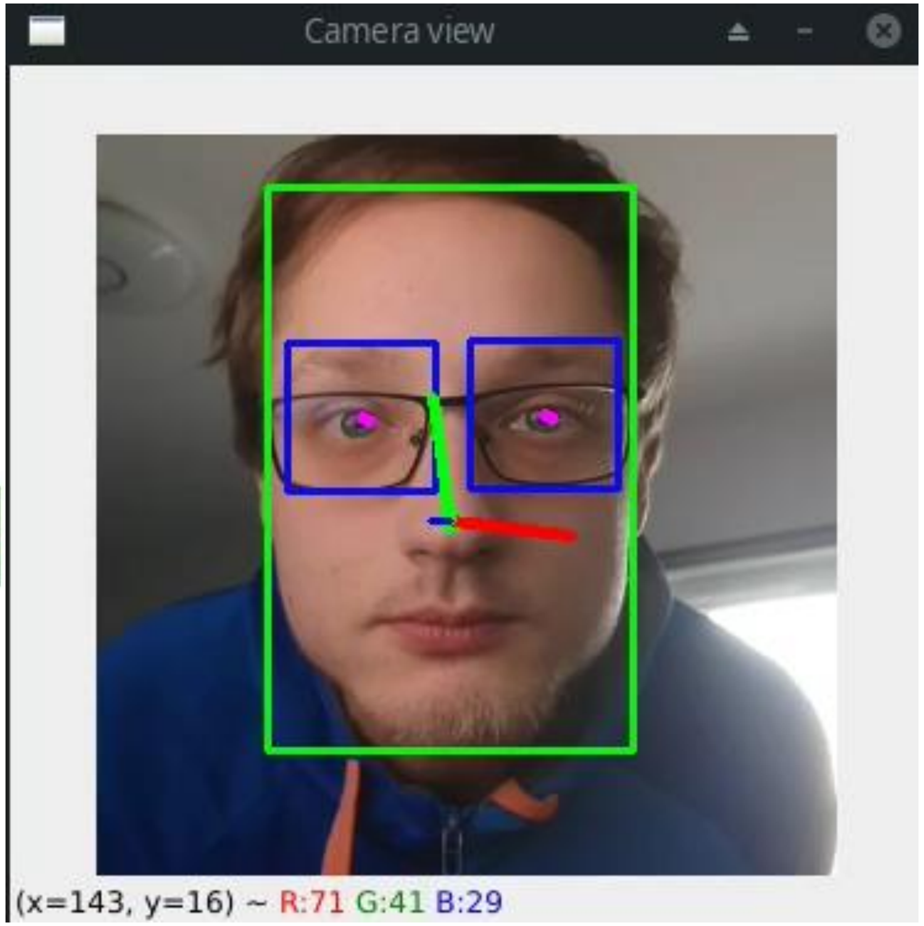
- Node and Graph Based Pipeline
- 3 Node Modalities:
  1. Pre-canned Hardware-accelerated CV/AI/Spatial functionalities
  2. CPython bindings for running scripts directly on DepthAI
  3. OpenCL, ML-Framework-based vectorized math (e.g here) for custom hardware-accelerated CV/AI/Spatial functionalities
- The network on chip builds the graph - allowing extremely high data-rate and low-latency connection between the nodes
- The DepthAI resource manager configures the network on chip

Gen2 Pipeline Builder Pre-Canned Nodes

- Neural inference
- 3D object localization
- Object tracking
- Stereo depth
- h.264/h.265 encoding
- background subtraction
- feature tracking
- motion estimation
- arbitrary crop/rescale/reformat and ROI return (e.g. allowing lossless zoom)

DepthAI Pipeline Builder Example


Input Image
Face detection model
face-detection-retail-0004
Cropped Face
Cropped Face
Landmark Détection Model
landmarks-regression-retail-0009
Head Pose Estimation Model
head-pose-estimation-adas-0001
Cropped Left and Right Eye
Head Pose Angles
Gaze Estimation Model
gaze-estimation-adas-0002
Gaze Coordinates



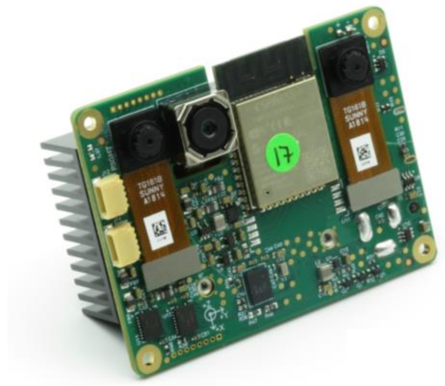
https://github.com/luxonis/depthai-experiments/tree/master/gen2-gaze-estimation#gen2-gaze-estimation

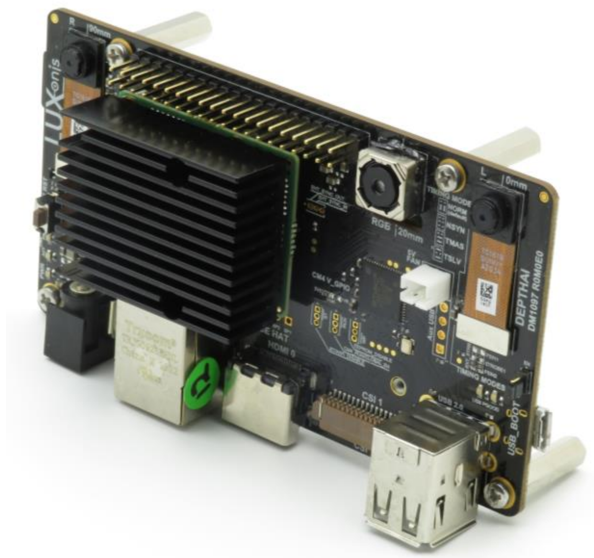
Built for Products – Open-Source, MIT-Licensed

- We built the platform as we, engineers, would want it:
  - Open Source so that it can be autonomously integrated into other codebases
  - Permissively licensed; it can be built into closed-source systems without concern
- DepthAI is Open-Source and MIT-Licensed
  - Hardware
  - Firmware
  - Software
  - ML-Training & Resources

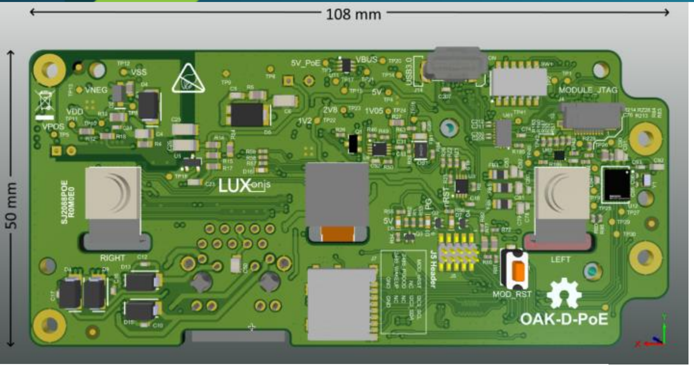
Open-Source Hardware – That You Can Buy

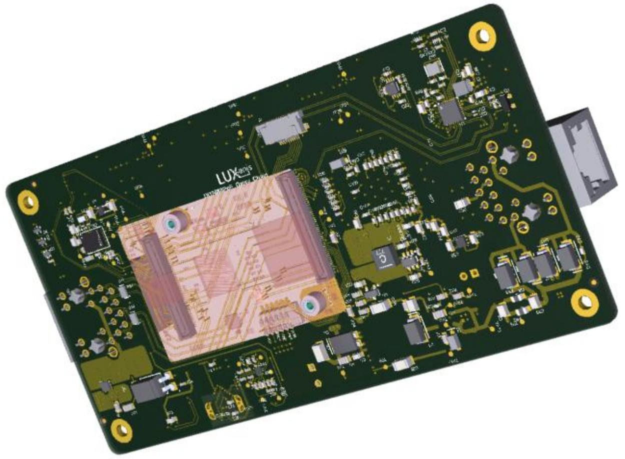
- Can buy all of these directly and us as supported products
- They are also all open-source reference designs, with full Altium Designer source files

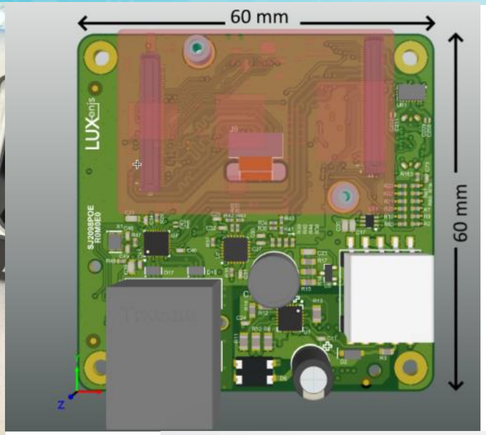


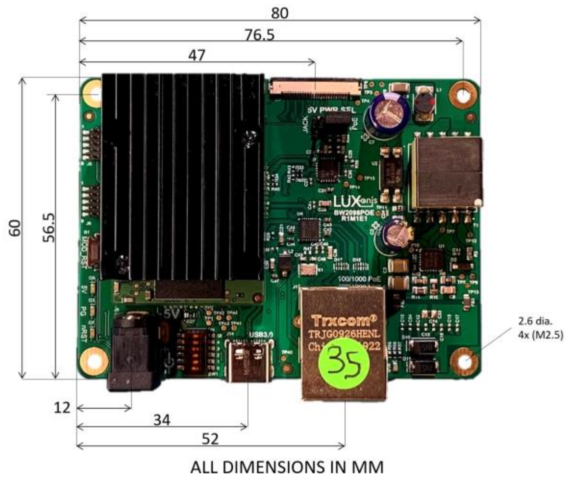


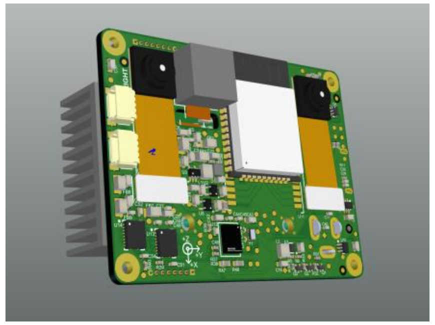
Open-Source Hardware

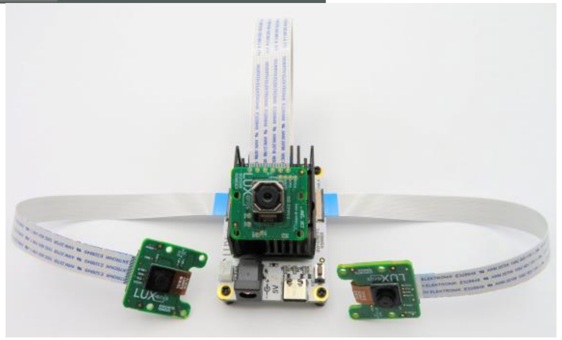


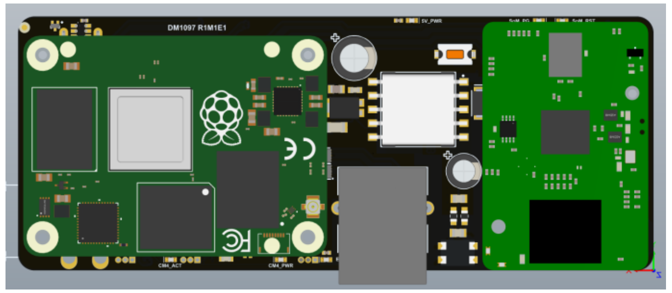












Open-Source Firmware

- Full DepthAI Pipeline Builder Available over SPI Interface, with C++:

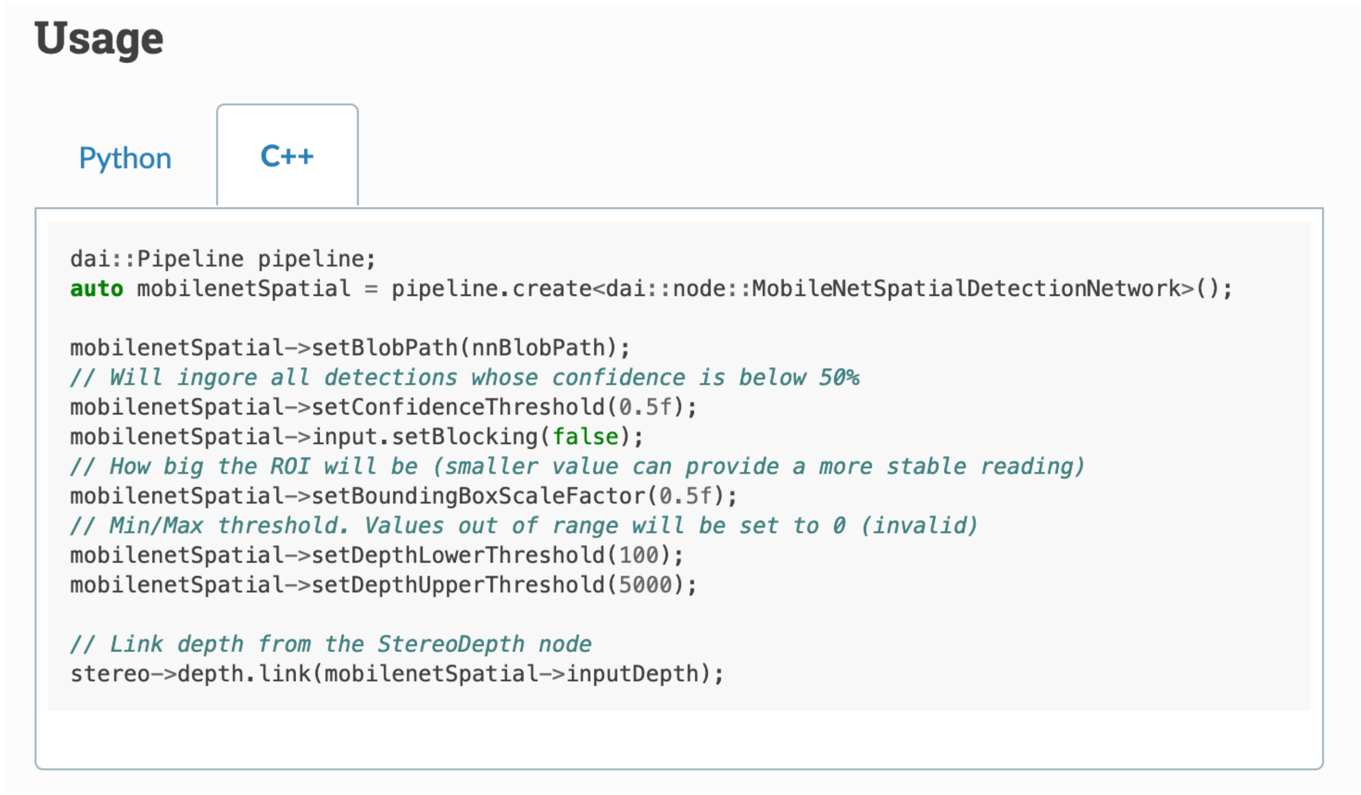
```
dai::Pipeline p;

// set up NN node
auto nn1 = p.create<dai::node::NeuralNetwork>();
nn1->setBlobPath(nnPath);

// set up color camera and link to NN node
auto colorCam = p.create<dai::node::ColorCamera>();
colorCam->setPreviewSize(300, 300);
colorCam->setResolution(dai::ColorCameraProperties::SensorResolution::THE_1080_P);
colorCam->setInterleaved(false);
colorCam->setCamId(0);
colorCam->setColorOrder(dai::ColorCameraProperties::ColorOrder::BGR);
colorCam->preview.link(nn1->input);

// set up SPI out node and link to nn1
auto spiOut = p.create<dai::node::SPIOut>();
spiOut->setStreamName("spimetaout");
spiOut->setBusId(0);
nn1->out.link(spiOut->input);

return p;
```

- This means you can no-joke have tinyYOLOv4 running at 30FPS with an ATmega8 “host”.
- ESP32, STM32, MSP430, etc. are commonly used.
- Others are easy to integrate
- microROS (ESP32) example

Open-Source Software

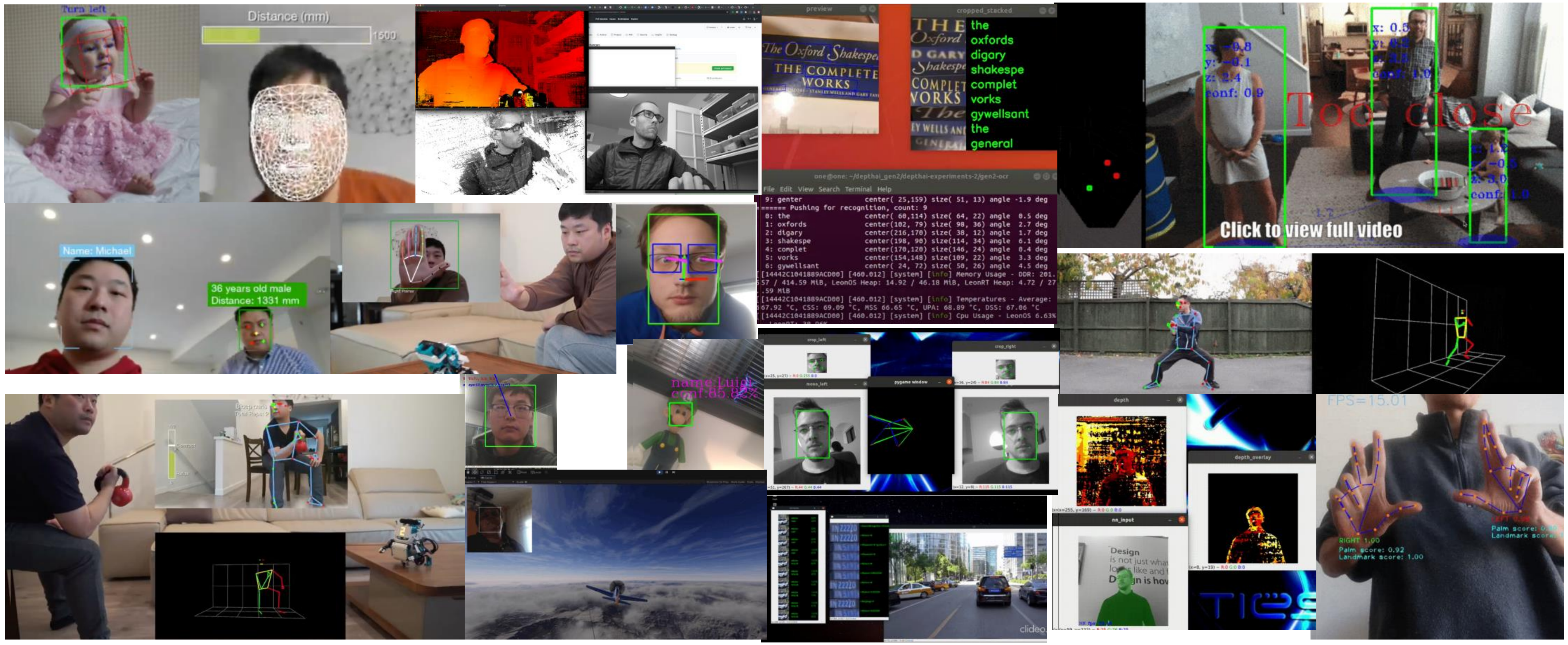
- Permissively (MIT-) Licensed so that closed-source products can be built royalty-free.
- We built this how we would want it.

- Python and C++ API parity
- ROS1 and ROS2 Integration
- Unity Plugin
- Wealth of reference pipelines



```cpp
dai::Pipeline pipeline;
auto mobilenetSpatial = pipeline.create<dai::node::MobileNetSpatialDetectionNetwork>();

mobilenetSpatial->setBlobPath(nnBlobPath);
// Will ingore all detections whose confidence is below 50%
mobilenetSpatial->setConfidenceThreshold(0.5f);
mobilenetSpatial->input.setBlocking(false);
// How big the ROI will be (smaller value can provide a more stable reading)
mobilenetSpatial->setBoundingBoxScaleFactor(0.5f);
// Min/Max threshold. Values out of range will be set to 0 (invalid)
mobilenetSpatial->setDepthLowerThreshold(100);
mobilenetSpatial->setDepthUpperThreshold(5000);

// Link depth from the StereoDepth node
stereo->depth.link(mobilenetSpatial->inputDepth);
```

Open-Source Software



```
one@one: ~/depthai_gen2/depthai-experiments-2/gen2-ocr
File  Edit  View  Search  Terminal  Help
 9: genter               center( 25,159) size( 51, 13) angle -1.9 deg
======= Pushing for recognition, count: 9
 0: the                  center( 60,114) size( 64, 22) angle  0.5 deg
 1: oxfords              center(102, 79) size( 98, 36) angle  2.7 deg
 2: digary               center(216,170) size( 38, 12) angle  1.7 deg
 3: shakespe             center(198, 90) size(114, 34) angle  6.1 deg
 4: complet              center(170,120) size(146, 24) angle  0.4 deg
 5: vorks                center(154,148) size(109, 22) angle  3.3 deg
 6: gywellsant           center( 24, 72) size( 50, 26) angle  4.5 deg
[14442C1041889ACD00] [460.012] [system] [info] Memory Usage - DDR: 201.
57 / 414.59 MiB, LeonOS Heap: 14.92 / 46.18 MiB, LeonRT Heap: 4.72 / 27
.59 MiB
[14442C1041889ACD00] [460.012] [system] [info] Temperatures - Average:
67.92 °C, CSS: 69.09 °C, MSS 66.65 °C, UPA: 68.89 °C, DSS: 67.06 °C
[14442C1041889ACD00] [460.012] [system] [info] Cpu Usage - LeonOS 6.63%
```

LUXonis

DepthAI Resources

DepthAI Resources

DepthAI Documentation

https://docs.luxonis.com/en/latest/

DepthAI Community Discord

https://discord.gg/EPsZHkg9Nx

Where to buy DepthAI:

- Mouser
- Sparkfun
- OpenCV

2021 Embedded Vision Summit

Luxonis Demos:

- From-Behind Collision Detection for People Who Ride Bikes
- Neural-Inference-Controlled Crop/Zoom and H.265 Encode
- Spatial AI and CV for Human Machine Safety