The International Workshop on Edge Artificial Intelligence for Industrial Applications (EAI4IA)

AI ECSEL JU

ANDANTE

AI4DI

TEMPO

Disruptive technology development through SMEs

A view from Neuromorphic edge AI

Dylan Muir SynSense Switzerland

Vienna, Austria 25-26 July 2022

SynSense

- ML inference ASICs
- Stateful NNs
- Asynchronous event-driven
- Single-bit sparse communication

Strategies for Neuromorphic ML

No “killer app” yet for Neuromorphic ML

Strategies for Neuromorphic ML

No “killer app” yet for Neuromorphic ML

→ Low-hanging fruit,
not cool apps

Strategies for Neuromorphic ML

Non-standard programming model

Strategies for Neuromorphic ML

SynSense

Non-standard programming model

→ Build programming methods

→ Work with early adopters

Strategies for Neuromorphic ML

SynSense

Steep learning curve for developers

Strategies for Neuromorphic ML

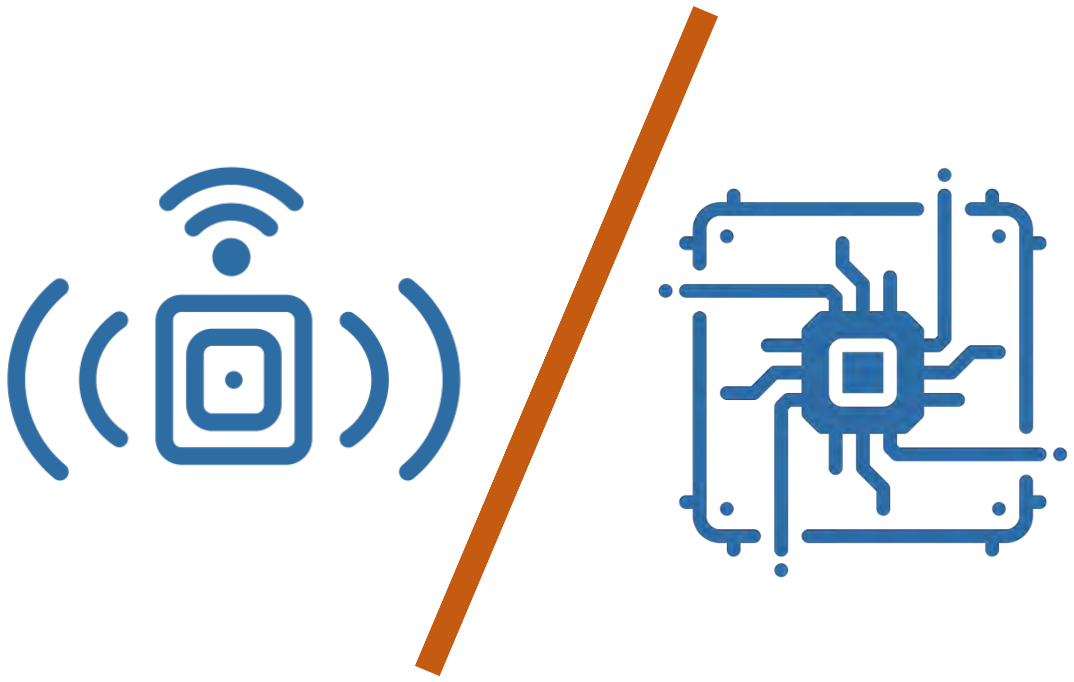
SynSense

Steep learning curve for developers

→ Build open SW pipelines to enable 3rd party development

Strategies for Neuromorphic ML

Traditionally siloed tech development



Strategies for Neuromorphic ML

SynSense

Traditionally siloed tech development

→ Full stack development

→ Custom sensor interfaces

→ Flexible architectures, specific use cases

Strategies for Neuromorphic ML

Skepticism from ML community

Strategies for Neuromorphic ML

SynSense

Skepticism from ML community

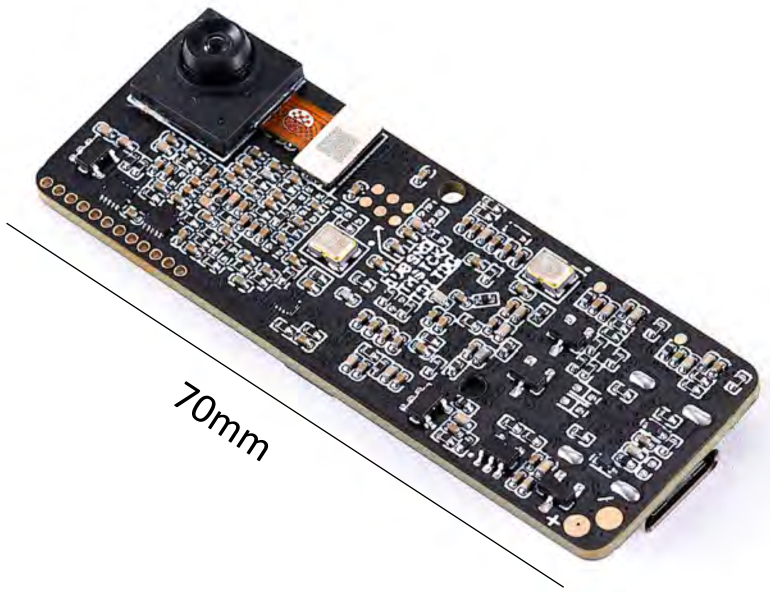
→ Show we can do it

→ Engage with research community

→ Engage with TinyML industry

Speck™ — Low-power integrated vision sensing

- Highly integrated SoC with vision sensor and processor on die
- Very compact module form-factor (10mm^2)
- CNN-based event-driven vision processing
- Low-power continuous operation (<5mW)



Design, training and deployment

Integrated SW toolchain

Import model conversion utils

```
from sinabs.from_torch import from_model
from sinabs.backend.dynapcnn import DynapcnnCompatibleNetwork
```

Convert model to a spiking CNN

```
spiking_model = from_model(cnn, input_shape).spiking_model
```

Map the model to DynapCNN processor cores …

```
dynapcnn_net = DynapcnnCompatibleNetwork(
   spiking_model,
   input_shape,
)
```

Create a hardware configuration

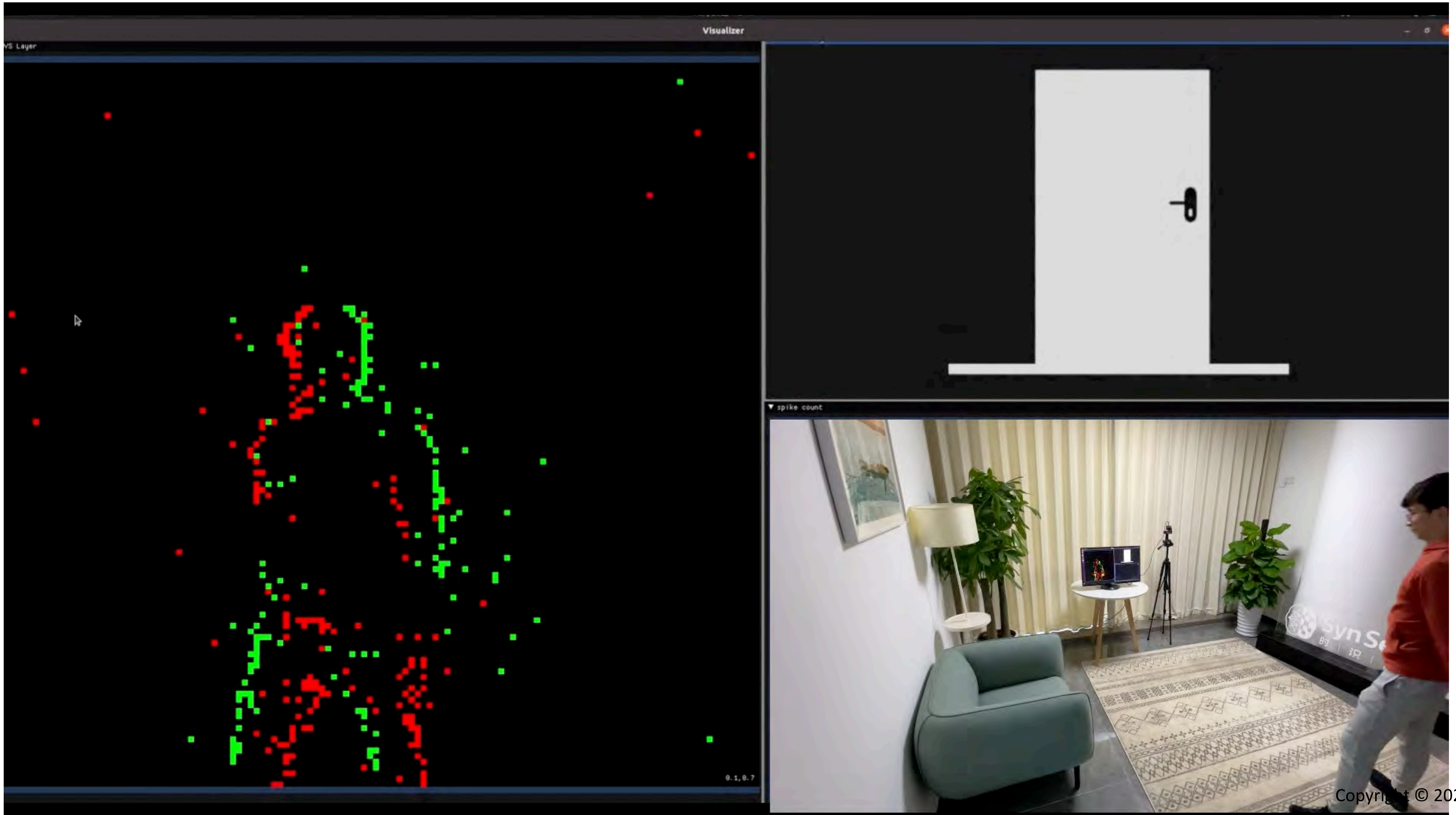
```
config = dynapcnn_net.make_config(device = "speck2b")
```

Send the configuration to the HDK

```
speck_hdk.apply_configuration(config)
```

sinabs.ai

Speck™ — Low-power integrated vision sensing



Thank You

For your attention

@ dylan.muir@synsense.ai