Autonomous Nano-UAVs: An Extreme Edge Computing Case

Daniele Palossi
daniele.palossi@idsia.ch

EMEA Innovation Forum 2022, October 10-12, Limassol CY
Motivation/Vision

Huge interest in autonomous (i.e., **no external infrastructure**) unmanned aerial vehicles (UAVs)
- Plethora of relevant applications in civil and military use cases
- Significant social and economic impact

- Precise agriculture
- Surveillance & Inspection
- Rescue missions
- Entertainment
Huge interest in autonomous (i.e., no external infrastructure) unmanned aerial vehicles (UAVs)

- Plethora of relevant applications in civil and military use cases
- Significant social and economic impact

**But size does matter!**

- Enhanced safety, e.g., Human Robot Interaction (HRI)
- Reduced costs
- New use cases:
  - Unreachable locations, ubiquitous Internet-of-Things, etc.
State-of-the-Art and Challenges

<table>
<thead>
<tr>
<th>Vehicle class</th>
<th>Ø : Weight [cm:Kg]</th>
<th>Power [W]</th>
<th>Onboard Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard-size</td>
<td>≥ 50 : ≥ 1</td>
<td>≥ 100</td>
<td>Desktop</td>
</tr>
<tr>
<td>micro-size</td>
<td>~25 : ~0.5</td>
<td>~50</td>
<td>Embedded</td>
</tr>
<tr>
<td>nano-size</td>
<td>~10 : ~0.01</td>
<td>~5</td>
<td>MCU</td>
</tr>
<tr>
<td>pico-size</td>
<td>≤ 2 : ≤ 0.001</td>
<td>≤ 0.1</td>
<td>ULP</td>
</tr>
</tbody>
</table>

Standard/Micro-sized UAVs:
- Autonomous driving, Racing, Agile/Acrobatic maneuvers.
- Simultaneous localization and mapping (SLAM).
- Optimal trajectory planning.

DJI MATRICE 300 RTK
State-of-the-Art and Challenges

<table>
<thead>
<tr>
<th>Vehicle class</th>
<th>Ø : Weight [cm:Kg]</th>
<th>Power [W]</th>
<th>Onboard Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard-size</td>
<td>≥ 50 : ≥ 1</td>
<td>≥ 100</td>
<td>Desktop</td>
</tr>
<tr>
<td>micro-size</td>
<td>~25 : ~0.5</td>
<td>~50</td>
<td>Embedded</td>
</tr>
<tr>
<td>nano-size</td>
<td>~10 : ~0.01</td>
<td>~5</td>
<td>MCU</td>
</tr>
<tr>
<td>pico-size</td>
<td>≤ 2 : ≤ 0.001</td>
<td>≤ 0.1</td>
<td>ULP</td>
</tr>
</tbody>
</table>

Standard/Micro-sized UAVs:
- Autonomous driving, Racing, Agile/Acrobatic maneuvers.
- Simultaneous localization and mapping (SLAM).
- Optimal trajectory planning.
State-of-the-Art and Challenges

<table>
<thead>
<tr>
<th>Vehicle class</th>
<th>Ø : Weight [cm:Kg]</th>
<th>Power [W]</th>
<th>Onboard Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard-size</td>
<td>≥ 50 : ≥ 1</td>
<td>≥ 100</td>
<td>Desktop</td>
</tr>
<tr>
<td>micro-size</td>
<td>~25 : ~0.5</td>
<td>~50</td>
<td>Embedded</td>
</tr>
<tr>
<td>nano-size</td>
<td>~10 : ~0.01</td>
<td>~5</td>
<td>MCU</td>
</tr>
<tr>
<td>pico-size</td>
<td>≤ 2 : ≤ 0.001</td>
<td>≤ 0.1</td>
<td>ULP</td>
</tr>
</tbody>
</table>

Standard/Micro-sized UAVs:
- Autonomous driving, Racing, Agile/Acrobatic maneuvers.
- Simultaneous localization and mapping (SLAM).
- Optimal trajectory planning.

*Automatic* vision-based remotely controlled UAVs are not sufficient in many scenarios, due to:
- Range
- Bandwidth
- Reliability
- Security
- Latency
- Power
- Range
- Bandwidth
- Reliability
- Security
- Latency
- Power
State-of-the-Art and Challenges

<table>
<thead>
<tr>
<th>Vehicle class</th>
<th>⌀: Weight [cm:Kg]</th>
<th>Power [W]</th>
<th>Onboard Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard-size</td>
<td>≥ 50 : ≥ 1</td>
<td>≥ 100</td>
<td>Desktop</td>
</tr>
<tr>
<td>micro-size</td>
<td>~25 : ~0.5</td>
<td>~50</td>
<td>Embedded</td>
</tr>
<tr>
<td>nano-size</td>
<td>~10 : ~0.01</td>
<td>~5</td>
<td>MCU</td>
</tr>
<tr>
<td>pico-size</td>
<td>≤ 2 : ≤ 0.001</td>
<td>≤ 0.1</td>
<td>ULP</td>
</tr>
</tbody>
</table>

Onboard computation budget only ~5% of total power.

**Challenge**: enable high-level intelligence aboard nano-size UAVs.

*R. J. Wood et al., “Progress on ‘pico’ air vehicles”, 2017.*
State-of-the-Art and Challenges

<table>
<thead>
<tr>
<th>Vehicle class</th>
<th>Ø : Weight [cm:Kg]</th>
<th>Power [W]</th>
<th>Onboard Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard-size</td>
<td>≥ 50 : ≥ 1</td>
<td>≥ 100</td>
<td>Desktop</td>
</tr>
<tr>
<td>micro-size</td>
<td>~25 : ~0.5</td>
<td>~50</td>
<td>Embedded</td>
</tr>
<tr>
<td>nano-size</td>
<td>~10 : ~0.01</td>
<td>~5</td>
<td>MCU</td>
</tr>
<tr>
<td>pico-size</td>
<td>≤ 2 : ≤ 0.001</td>
<td>≤ 0.1</td>
<td>ULP</td>
</tr>
</tbody>
</table>

Onboard computation budget only ~5% of total power. **Challenge**: enable high-level intelligence aboard nano-size UAVs.

Nano-UAVs show a challenging trade-off between:
- power consumption, i.e., ≤ ~200 mW
- real-time onboard processing, e.g., ≥ ~10 frame/s

---

State-of-the-Art and Challenges

<table>
<thead>
<tr>
<th>Vehicle class</th>
<th>Ø : Weight [cm:Kg]</th>
<th>Power [W]</th>
<th>Onboard Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard-size</td>
<td>≥ 50 : ≥ 1</td>
<td>≥ 100</td>
<td>Desktop</td>
</tr>
<tr>
<td>micro-size</td>
<td>~25 : ~0.5</td>
<td>~50</td>
<td>Embedded</td>
</tr>
<tr>
<td>nano-size</td>
<td>~10 : ~0.01</td>
<td>~5</td>
<td>MCU</td>
</tr>
<tr>
<td>pico-size</td>
<td>≤ 2 : ≤ 0.001</td>
<td>≤ 0.1</td>
<td>ULP</td>
</tr>
</tbody>
</table>

Onboard computation budget only ~5% of total power.
**Challenge**: enable high-level intelligence aboard nano-size UAVs.

Nano-UAVs show a challenging trade-off between:
- power consumption, i.e., ≤ ~200 mW
- real-time onboard processing, e.g., ≥ ~10 frame/s

**Extreme edge computing!**

Can we achieve ~10 frame/s in ~200 mW within few grams payload? How to get such a high energy efficiency?

How to enable the extreme edge computing?


D. Palossi
How to enable the extreme edge computing?


How to enable the extreme edge computing?

ULP Heterogeneous Model [1]

Parallel Execution [2]

Approximate Computing [3]

**Question:**
10.11 × 9.8 ?

**Answer:**
About 100

In our context:
- missing FPUs (hardware)
- floating point ➡ fixed-point arithmetic (quantization)

---


How to enable the extreme edge computing?

**ULP Heterogeneous Model [1]**

**Parallel Execution [2]**

**Approximate Computing [3]**

**Deep learning [3]**

---

**Question:**
10.11 \times 9.8 ?

**Answer:**
About 100

In our context:
- missing FPUs (hardware)
- floating point $\rightarrow$ fixed-point arithmetic (quantization)

Deep learning (DL):
- e.g., *end-to-end learning.*
- can be “lighter” than traditional workloads, such as the *localization mapping-planning cycle.*

---

Full stack application: Autonomous navigation

Dronet: learning to fly by driving [4]

Full stack application: Autonomous navigation

Dronet: learning to fly by driving [4]

- CNN based on RES-Net topology
- Indoor/outdoor lane following
- Obstacle avoidance
- Off-board processing @ 20fps

<table>
<thead>
<tr>
<th>Output</th>
<th>Training Dataset</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steering angle (regression)</td>
<td>Udacity (self-driving car)</td>
<td>EVA 0.737</td>
</tr>
<tr>
<td>Collision probability (classification)</td>
<td>Zürich bicycle</td>
<td>Avg. Acc. 95.4%</td>
</tr>
</tbody>
</table>

Full stack application: Autonomous navigation

Dronet: learning to fly by driving [4]

- CNN based on RES-Net topology
- Indoor/outdoor lane following
- Obstacle avoidance
- Off-board processing @ 20fps

- 41MMAC/frame → 1GOP/s @ 10fps
- 3.2×10^5 parameters > 1MB
- Intel i7 → tens of Watts

<table>
<thead>
<tr>
<th>Output</th>
<th>Training Dataset</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steering angle (regression)</td>
<td>Udacity (self-driving car)</td>
<td>EVA 0.737</td>
</tr>
<tr>
<td>Collision probability (classification)</td>
<td>Zürich bicycle</td>
<td>Avg. Acc. 95.4%</td>
</tr>
</tbody>
</table>

On-board brain
On-board brain

GAP8 architecture

We need to handle:
- Memory management
- Float ➡ Fixed-point
- Architectural parallelism

Tedious and error-prone manual coding.
Incremental modifications of the Dronet [4]:

- Batch Normalization Folding
  - Direct and inverse folding due to the by-pass topology
- Fixed-point 16 bit instead of Float 32 bit
  - Signed Q5.11 directly in training
- Max pooling 2×2 instead of 3×3
  - Minimal loss in accuracy with 2.25× reduction time
- Training dataset extension
  - ~1300 collision images with the same onboard camera
PULP-Dronet: Shrinking and optimization

Incremental modifications of the Dronet [4]:

- Batch Normalization Folding
  - Direct and inverse folding due to the by-pass topology
- Fixed-point 16 bit instead of Float 32 bit
  - Signed Q5.11 directly in training
- Max pooling 2×2 instead of 3×3
  - Minimal loss in accuracy with 2.25× reduction time
- Training dataset extension
  - ~1300 collision images with the same onboard camera

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Max pooling</th>
<th>Data Type</th>
<th>Regression (steering)</th>
<th>Classification (collision)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>3 × 3</td>
<td>Float32</td>
<td>0.109</td>
<td>0.952</td>
</tr>
<tr>
<td>Extended</td>
<td>2 × 2</td>
<td>Fixed16</td>
<td>0.110</td>
<td>0.977</td>
</tr>
</tbody>
</table>
PULP-Dronet: Shrinking and optimization

Incremental modifications of the Dronet [4]:

- Batch Normalization Folding
  - Direct and inverse folding due to the by-pass topology
- Fixed-point 16 bit instead of Float 32 bit
  - Signed Q5.11 directly in training
- Max pooling 2×2 instead of 3×3
  - Minimal loss in accuracy with 2.25× reduction time
- Training dataset extension
  - ~1300 collision images with the same onboard camera

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Max pooling</th>
<th>Data Type</th>
<th>Training</th>
<th>Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>RMSE</td>
<td>Accuracy</td>
</tr>
<tr>
<td>Original</td>
<td>3 × 3</td>
<td>Float32</td>
<td>0.109</td>
<td>0.952</td>
</tr>
<tr>
<td>Extended</td>
<td>2 × 2</td>
<td>Fixed16</td>
<td>0.110</td>
<td>0.977</td>
</tr>
</tbody>
</table>

Maximize the parallel computation:

- **Spatial scheme** (horizontal)
  - Requires sufficient amount of parallel work within the same tile
  - Used for the 1st layer (conv + max pooling)
- **Feature-wise scheme**
  - Heavier memory footprint + reduction stage
  - Used for all the remaining layers
Sweeping between several operating modes on GAP8:

- VDD: 1.0-1.2V
- Fabric Ctrl (FC) / Cluster (CL): 25-250MHz**

**beyond GAP8 official spec: https://gwt-website-files.s3.amazonaws.com/gap8_datasheet.pdf
Sweeping between several operating modes on GAP8:

- VDD: 1.0-1.2V
- Fabric Ctrl (FC) / Cluster (CL): 25-250MHz**

Most energy-efficient configuration:
VDD@1.0V, FC@50MHz, CL@100MHz

**beyond GAP8 official spec: https://gwt-website-files.s3.amazonaws.com/gap8_datasheet.pdf
Sweeping between several operating modes on GAP8:

- VDD: 1.0-1.2V
- Fabric Ctrl (FC) / Cluster (CL): 25-250MHz**

** Most energy-efficient configuration
VDD@1.0V, FC@50MHz, CL@100MHz

** beyond GAP8 official spec: https://gwt-website-files.s3.amazonaws.com/gap8_datasheet.pdf
PULP-Dronet: Results

Sweeping between several operating modes on GAP8:

- VDD: 1.0-1.2V
- Fabric Ctrl (FC) / Cluster (CL): 25-250MHz**

Most energy-efficient configuration:
VDD@1.0V, FC@50MHz, CL@100MHz

6fps @ 45mW
VDD@1.0V, FC@50MHz, CL@100MHz

18fps @ 272mW
VDD@1.2V, FC@250MHz, CL@250MHz

**beyond GAP8 official spec: https://gwt-website-files.s3.amazonaws.com/gap8_datasheet.pdf
PULP-Dronet: Results

- DRAM active only for transferring time
- PULP-Shield + Crazyflie electronics:
  - Most energy efficient: 4.4% of total power (64mW)
  - Highest performance: 7% of total power (284mW)
- Frame-rate:
  - Most energy efficient: 6fps
  - Highest performance: 18fps
PULP-Dronet: Results

Power envelope break-down of the entire cyber-physical system:

### Lifetime cost:
- 20% → weight
- 2% → power consumption

<table>
<thead>
<tr>
<th></th>
<th>Crazyflie 2.0</th>
<th>Crazyflie 2.0 + PULP-Shield</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lifetime [m:s]</strong></td>
<td>~7:20</td>
<td>~5:50</td>
</tr>
<tr>
<td>PULP-Shield Off</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PULP-Shield On</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

GAP8 VDD@1.0V, FC@50MHz, CL@100MHz

- DRAM active only for transferring time
- PULP-Shield + Crazyflie electronics:
  - Most energy efficient: 4.4% of total power (64mW)
  - Highest performance: 7% of total power (284mW)
- Frame-rate:
  - Most energy efficient: 6fps
  - Highest performance: 18fps
PULP-Dronet: In-field evaluation

Longest indoor traveled distance → ~113m

Collision avoidance capability
- Straight flight @ 1.5 m/s
- Dynamic obstacle appears at 2 meters
- Only 2 meters to brake
### PULP-Dronet v1

<table>
<thead>
<tr>
<th><strong>Topology:</strong></th>
<th>RESNet-based (BN-ReLU-Conv)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dataset:</strong></td>
<td>Udacity + Zürich bicycle + Himax</td>
</tr>
<tr>
<td><strong>Quantization:</strong></td>
<td>16bit</td>
</tr>
<tr>
<td><strong>Model size:</strong></td>
<td>320k parameters → 640kB</td>
</tr>
<tr>
<td><strong>Device:</strong></td>
<td>PULP-shield (PULP GAP8)</td>
</tr>
<tr>
<td><strong>Framework:</strong></td>
<td>TensorFlow</td>
</tr>
<tr>
<td><strong>Tools:</strong></td>
<td>Custom scripts + GWT AutoTiler</td>
</tr>
<tr>
<td><strong>Performance:</strong></td>
<td>6 fps @ 45 mW (energy efficient) 18 fps @ 272 mW (max performance)</td>
</tr>
<tr>
<td><strong>Publications:</strong></td>
<td>10.1109/JIOT.2019.2917066 10.1109/DCOSS.2019.00111</td>
</tr>
<tr>
<td><strong>GitHub</strong></td>
<td><a href="https://github.com/pulp-platform/pulp-dronet/tree/master/pulp-dronet-v1">https://github.com/pulp-platform/pulp-dronet/tree/master/pulp-dronet-v1</a></td>
</tr>
</tbody>
</table>

**Limitations:**
- Memory footprint (640kB)
- Custom PCB
- Quantization and Deployment partially hand-crafted
## PULP-Dronet evolution

<table>
<thead>
<tr>
<th>Topology:</th>
<th>RESNet-based (BN-ReLU-Conv)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset:</td>
<td>Udacity + Zürich bicycle + Himax</td>
</tr>
<tr>
<td>Quantization:</td>
<td>16bit</td>
</tr>
<tr>
<td>Model size:</td>
<td>320k parameters → 640kB</td>
</tr>
<tr>
<td>Device:</td>
<td>PULP-shield (PULP GAP8)</td>
</tr>
<tr>
<td>Framework:</td>
<td>TensorFlow</td>
</tr>
<tr>
<td>Tools:</td>
<td>Custom scripts + GWT AutoTiler</td>
</tr>
</tbody>
</table>
| Performance: | 6 fps @ 45 mW (energy efficient)  
18 fps @ 272 mW (max performance) |
| Publications: | 10.1109/JIOT.2019.2917066  
10.1109/DCOSS.2019.00111 |
| GitHub | https://github.com/pulp-platform/pulp-dronet/tree/master/pulp-dronet-v1 |

### Limitations:
- Memory footprint (640kB)
- Custom PCB
- Quantization and Deployment partially hand-crafted

### New features:
- 8bit quantization
- Commercial PCB
- Automated Quantization and Deployment (PyTorch)
# PULP-Dronet v2

<table>
<thead>
<tr>
<th>Topology:</th>
<th>PULP-Dronet v1</th>
<th>PULP-Dronet v2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RESNet-based <em>(BN-ReLU-Conv)</em></td>
<td>RESNet-based <em>(Conv-BN-ReLU)</em></td>
</tr>
<tr>
<td>Dataset:</td>
<td>Udacity + Zürich bicycle + Himax</td>
<td>Udacity + Zürich bicycle + Himax</td>
</tr>
<tr>
<td>Quantization:</td>
<td>16bit</td>
<td>8bit</td>
</tr>
<tr>
<td>Model size:</td>
<td>320k parameters ➔ 640kB</td>
<td>320k parameters ➔ 320kB</td>
</tr>
<tr>
<td>Device:</td>
<td>PULP-shield (PULP GAP8)</td>
<td>AI-deck (PULP GAP8)</td>
</tr>
<tr>
<td>Framework:</td>
<td>TensorFlow</td>
<td>PyTorch</td>
</tr>
<tr>
<td>Performance:</td>
<td>6 fps @ 45 mW <em>(energy efficient)</em></td>
<td>9 fps @ 40 mW</td>
</tr>
<tr>
<td></td>
<td>18 fps @ 272 mW <em>(max performance)</em></td>
<td>17 fps @ 119 mW</td>
</tr>
<tr>
<td>Publications:</td>
<td>● 10.1109/JIOT.2019.2917066</td>
<td>● 10.1109/AICAS51828.2021.9458550</td>
</tr>
<tr>
<td></td>
<td>● 10.1109/DCOSS.2019.00111</td>
<td>● 10.1109/JETCAS.2021.3126259</td>
</tr>
</tbody>
</table>

---

PULP-Dronet v2: Results

Regression/Classification performances comparable with PULP-Dronet v1 despite the stronger 8bit quantization.
Regression/Classification performances comparable with PULP-Dronet v1 despite the stronger 8bit quantization.
## PULP-Dronet evolution

<table>
<thead>
<tr>
<th></th>
<th>PULP-Dronet v1</th>
<th>PULP-Dronet v2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Topology:</strong></td>
<td>RESNet-based (BN-ReLU-Conv)</td>
<td>RESNet-based (Conv-BN-ReLU)</td>
</tr>
<tr>
<td><strong>Dataset:</strong></td>
<td>Udacity + Zürich bicycle + Himax</td>
<td>Udacity + Zürich bicycle + Himax</td>
</tr>
<tr>
<td><strong>Quantization:</strong></td>
<td>16bit</td>
<td>8bit</td>
</tr>
<tr>
<td><strong>Model size:</strong></td>
<td>320k parameters ➡ 640kB</td>
<td>320k parameters ➡ 320kB</td>
</tr>
<tr>
<td><strong>Device:</strong></td>
<td>PULP-shield (PULP GAP8)</td>
<td>AI-deck (PULP GAP8)</td>
</tr>
<tr>
<td><strong>Framework:</strong></td>
<td>TensorFlow</td>
<td>PyTorch</td>
</tr>
<tr>
<td><strong>Tools:</strong></td>
<td>Custom scripts + GWT AutoTiler</td>
<td>GWT GAPflow [4] v1.0.0</td>
</tr>
<tr>
<td><strong>Performance:</strong></td>
<td>6 fps @ 45 mW (energy efficient)</td>
<td>9 fps @ 40 mW</td>
</tr>
<tr>
<td></td>
<td>18 fps @ 272 mW (max performance)</td>
<td>17 fps @ 119 mW</td>
</tr>
<tr>
<td><strong>Publications:</strong></td>
<td>10.1109/JIOT.2019.2917066</td>
<td>10.1109/AICAS51828.2021.9458550</td>
</tr>
<tr>
<td></td>
<td>10.1109/DCOSS.2019.00111</td>
<td>10.1109/JETCAS.2021.3126259</td>
</tr>
</tbody>
</table>

### v2 limitation
- High computational cost still preventing efficient multi-tasking

---

PULP-Dronet v3: Tiny-PULP-Dronets

PULP-Dronet v1/v2
RESNet-like topology
PULP-Dronet v3: Tiny-PULP-Dronets

PULP-Dronet v1/v2
RESNet-like topology
PULP-Dronet v3: Tiny-PULP-Dronets

PULP-Dronet v1/v2
RESNet-like topology

Overfitting?

# channels exploration:
- Baseline: 32 - 64 - 128 (/1)
- 16 - 32 - 64 (/2)
- 8 - 16 - 32 (/4)
- Smallest: 4 - 8 - 16 (/8)
**PULP-Dronet v3: Tiny-PULP-Dronets**

PULP-Dronet v1/v2
RESNet-like topology

**Structural sparsity analysis**: how many neurons are always inactive (on the test set)?

**# channels exploration:**
- Baseline: 32 - 64 - 128 (/1)
- 16 - 32 - 64 (/2)
- 8 - 16 - 32 (/4)
- Smallest: 4 - 8 - 16 (/8)
PULP-Dronet v3: Tiny-PULP-Dronets

PULP-Dronet v1/v2
RESNet-like topology

Structural sparsity analysis: how many neurons are always inactive (on the test set)?

<table>
<thead>
<tr>
<th></th>
<th>Act0</th>
<th>Act1</th>
<th>Act2</th>
<th>Act3</th>
<th>Act4</th>
<th>Act5</th>
<th>Act6</th>
<th>Act7</th>
<th>Act8</th>
<th>Act9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>28%</td>
<td>13%</td>
<td>6%</td>
<td>6%</td>
<td>25%</td>
<td>10%</td>
<td>11%</td>
<td>49%</td>
<td>60%</td>
<td>92%</td>
</tr>
</tbody>
</table>

# channels exploration:
- Baseline: 32 - 64 - 128 (/1)
- 16 - 32 - 64 (/2)
- 8 - 16 - 32 (/4)
- Smallest: 4 - 8 - 16 (/8)
PULP-Dronet v3: Tiny-PULP-Dronets

PULP-Dronet v1/v2
RESNet-like topology

Act0
Act1
Act2
Act3
Act4
Act5
Act6
Act7
Act8
Act9

Structural sparsity analysis: how many neurons are always inactive (on the test set)?

<table>
<thead>
<tr>
<th></th>
<th>Act0</th>
<th>Act1</th>
<th>Act2</th>
<th>Act3</th>
<th>Act4</th>
<th>Act5</th>
<th>Act6</th>
<th>Act7</th>
<th>Act8</th>
<th>Act9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>28%</td>
<td>13%</td>
<td>6%</td>
<td>6%</td>
<td>25%</td>
<td>10%</td>
<td>11%</td>
<td>49%</td>
<td>60%</td>
<td>92%</td>
</tr>
</tbody>
</table>

Residuals on/off exploration

Smallest 0.15% 0% 0% --- 0% 0.07% --- 0% 0% ---

# channels exploration:

- Baseline: 32 - 64 - 128 (/1)
- 16 - 32 - 64 (/2)
- 8 - 16 - 32 (/4)
- Smallest: 4 - 8 - 16 (/8)
Tiny-PULP-Dronets: Results

Baseline (/1) PULP-Dronet v2:
- Size: 320kB
- MAC: 41M
Tiny-PULP-Dronets: Results

Baseline (/1) PULP-Dronet v2:
- **Size**: 320kB
- **MAC**: 41M

Tiny-PULP-Dronets (/2,/4,/8):
- **model shrinking**

![Graph showing comparison betweenBaseline v2 and Tiny-PULP-Dronets with size (kB) and MAC (M) metrics.](image-url)
Tiny-PULP-Dronets: Results

Baseline (/1) PULP-Dronet v2:
- **Size**: 320kB
- **MAC**: 41M

Tiny-PULP-Dronets (/2,/4,/8):
- model shrinking
- by-pass removal
Tiny-PULP-Dronets: Results

Baseline (/1) PULP-Dronet v2:
- Size: 320kB
- MAC: 41M

Tiny-PULP-Dronets (/2,/4,/8):
- model shrinking
- by-pass removal

Regression:
- RMSE +0.01 (at most)
Tiny-PULP-Dronets: Results

Baseline (/1) PULP-Dronet v2:
- Size: 320kB
- MAC: 41M

Tiny-PULP-Dronets (/2,/4,/8):
- model shrinking
- by-pass removal

Regression:
- RMSE +0.01 (at most)

Classification:
- Accuracy -4% (at most)
Tiny-PULP-Dronets: Results

Baseline (/1) PULP-Dronet v2:
- **Size**: 320kB
- **MAC**: 41M

Tiny-PULP-Dronets (/2,/4,/8):
- Model shrinking
- By-pass removal

Regression:
- **RMSE**: +0.01 (at most)

Classification:
- **Accuracy**: -4% (at most)

Smallest vs. Baseline
- 50x smaller
- 27x faster
- Small drop in Accuracy
# PULP-Dronet evolution

<table>
<thead>
<tr>
<th></th>
<th>PULP-Dronet v1</th>
<th>PULP-Dronet v2</th>
<th>PULP-Dronet v3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Topology:</strong></td>
<td>RESNet-based (BN-ReLU-Conv)</td>
<td>RESNet-based (Conv-BN-ReLU)</td>
<td>v2 w/o by-pass (Conv-BN-ReLU)</td>
</tr>
<tr>
<td><strong>Dataset:</strong></td>
<td>Udacity + Zürich bicycle + Himax</td>
<td>Udacity + Zürich bicycle + Himax</td>
<td>Udacity + Zürich bicycle + Himax</td>
</tr>
<tr>
<td><strong>Quantization:</strong></td>
<td>16bit</td>
<td>8bit</td>
<td>8bit</td>
</tr>
<tr>
<td><strong>Model size:</strong></td>
<td>320k parameters ➔ 640kB</td>
<td>320k parameters ➔ 320kB</td>
<td>320k -- 6k parameters ➔ 320kB -- 6kB</td>
</tr>
<tr>
<td><strong>Device:</strong></td>
<td>PULP-shield (PULP GAP8)</td>
<td>AI-deck (PULP GAP8)</td>
<td>AI-deck (PULP GAP8)</td>
</tr>
<tr>
<td><strong>Framework:</strong></td>
<td>TensorFlow</td>
<td>PyTorch</td>
<td>PyTorch</td>
</tr>
<tr>
<td><strong>Performance:</strong></td>
<td>6 fps @ 45 mW (energy efficient)</td>
<td>9 fps @ 40 mW</td>
<td>10 fps @ 34 mW</td>
</tr>
<tr>
<td></td>
<td>18 fps @ 272 mW (max performance)</td>
<td>17 fps @ 119 mW</td>
<td>19 fps @ 102 mW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10 fps @ 35 mW</td>
<td>10 fps @ 34 mW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>19 fps @ 102 mW</td>
<td>10 fps @ 34 mW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>160 fps @ 102 mW</td>
<td>89 fps @ 34 mW</td>
</tr>
<tr>
<td><strong>Publications:</strong></td>
<td>● 10.1109/JIOT.2019.2917066</td>
<td>● 10.1109/AICAS51828.2021.9458550</td>
<td>● 10.1109/AICAS54282.2022.9869931</td>
</tr>
<tr>
<td></td>
<td>● 10.1109/DCOSS.2019.00111</td>
<td>● 10.1109/JETCAS.2021.3126259</td>
<td></td>
</tr>
</tbody>
</table>


Full stack application: Human robot interaction


- RES-Net based
- 96.5 MMAC/frame
- > 6MB weights

PULP-Frontnet
- New dataset from the nano-drone camera
- Dataset augmentation (pitch, photometric)

<table>
<thead>
<tr>
<th></th>
<th>PULP-Frontnet 160x32</th>
<th>PULP-Frontnet 160x16</th>
<th>PULP-Frontnet 80x32</th>
<th>Proximity [6]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ops [MMAC]</td>
<td>14.1</td>
<td>4.3</td>
<td>4.0</td>
<td>96.5</td>
</tr>
<tr>
<td>Memory [kB]</td>
<td>499</td>
<td>184</td>
<td>348</td>
<td>6116</td>
</tr>
</tbody>
</table>

- CNN model scaling
- Model 8-bit quantization/deployment
Regression performance assessment $R^2$ score*:

- Perfect regressor $\rightarrow R^2 = 1.0$
- Dummy regressor (always predicting the mean) $\rightarrow R^2 = 0.0$

*R$^2$ is proportion of the variation in the dependent variable that is predictable from the independent variable(s)
PULP-Frontnet: Results

Regression performance assessment $R^2$ score*:
- Perfect regressor $\Rightarrow R^2 = 1.0$
- Dummy regressor (always predicting the mean) $\Rightarrow R^2 = 0.0$

- Three SoC configurations (voltage/frequencies)
  1. Peak throughput
  2. Most energy efficient
  3. Minimum power consumption
- Frame-rate:
- Max power: $160\times16$ 110.7fps @ 99mW (config. 1)
- Max efficiency: $80\times32$ 48fps @ 20mW (config. 2)
- Min power: $80\times32$ 18.5fps @ 8.6mW (config. 3)

\* $R^2$ is proportion of the variation in the dependent variable that is predictable from the independent variable(s)
PULP-Frontnet: In-field evaluation
How to improve the generalization capability?

Generalization problem:

- visual cues learned on the training domain hardly predict with the same accuracy on different ones;
- it limits the applicability/deployability of DNNs outside controlled environments;
- common to many DNNs application domains.
How to improve the generalization capability?

Generalization problem:

- visual cues learned on the training domain hardly predict with the same accuracy on different ones;
- it limits the applicability/deployability of DNNs outside controlled environments;
- common to many DNNs application domains.
How to improve the generalization capability?

Generalization problem:
- visual cues learned on the training domain hardly predict with the same accuracy on different ones;
- it limits the applicability/deployability of DNNs outside controlled environments;
- common to many DNNs application domains.

<table>
<thead>
<tr>
<th>Training domain</th>
<th>Deployment domain</th>
<th>Real world data</th>
<th>Simulated data</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Training domain" /></td>
<td><img src="image2.png" alt="Deployment domain" /></td>
<td><img src="image3.png" alt="Real world data" /></td>
<td><img src="image4.png" alt="Simulated data" /></td>
</tr>
</tbody>
</table>

- Expensive, time-consuming, and ad-hoc infrastructure

Need for realistic simulators (gap sim2real) including humans behaviours
How to improve the generalization capability?

Generalization problem:
- visual cues learned on the training domain hardly predict with the same accuracy on different ones;
- it limits the applicability/deployability of DNNs outside controlled environments;
- common to many DNNs application domains.

Training domain  Deployment domain
Real world data
Augmentation
Simulated data

Expensive, time-consuming, and ad-hoc infrastructure

Real-world data (existing dataset) + background perturbation

Need for realistic simulators (gap sim2real) including humans behaviours

Background randomization
Background randomization: Pipeline

Background randomization
Goal: teach the neural network to focus on the foreground person and ignore backgrounds

Background randomization: Pipeline

**Background randomization**
Goal: teach the neural network to focus on the foreground person and ignore backgrounds

Input image $\rightarrow$ Segmentation with Mask R-CNN [7] $\rightarrow$ Foreground person $\rightarrow$ Random background $\rightarrow$ Output image

Data augmentation is only applied at training time ➔ no impact on energy/performance on the inference aboard the nano-drone

Datasets are collected in two different indoor environments, subjects are disjoint between datasets.

Two models are trained on the training set from environment A.

We evaluate model performances on test sets from known environment A and never-seen-before environment B.

**Baseline:** PULP-Frontnet pipeline

**Proposed:** Background randomization pipeline
Background randomization: Results

- On the known test set (A) the baseline model performs better… but not on all output.
On the known test set (A) the baseline model performs better… but not on all output.

On the never-seen-before test set (B) the proposed model performs better (almost on all output variables).
Background randomization: Results

- On the known test set (A) the baseline model performs better… but not on all output.
- On the never-seen-before test set (B) the proposed model performs better (almost on all output variables).
- The drop of the baseline model is stronger than the one of the proposed model.
Background randomization: Results

- On the known test set (A) the baseline model performs better… but not on all output.
- On the never-seen-before test set (B) the proposed model performs better (almost on all output variables).
- The drop of the baseline model is stronger than the one of the proposed model.
- But on z both models have some issue – how to improve it?

Output variables

- Known test set A
  - On the known test set (A) the baseline model performs better… but not on all output.
  - On the never-seen-before test set (B) the proposed model performs better (almost on all output variables).
  - The drop of the baseline model is stronger than the one of the proposed model.
  - But on z both models have some issue – how to improve it?

- Never-seen-before test set B

---

D. Palossi 65
Background randomization: In-field evaluation

We test PULP-Frontnet with background randomization in the never-seen-before coffee room.
How to improve the regression performance?

Altitude/Pitch ambiguity:

Closed-loop drone’s pipeline:

- Visual input → Reactive CNN → Perception estimate → Low-level control
- Inertial inputs → State estimation
- Commands → Actuators

State estimation
How to improve the regression performance?

Altitude/Pitch ambiguity:

Closed-loop drone’s pipeline:

Provide the drone’s pitch as an explicit auxiliary input to the CNN, to facilitate its perception task:

PULP-Frontnet CNN with additional multi-layer perceptron branch for the state input
Vision-state fusion: Results

Test set regression performance, $R^2$ score (the higher the better) vs. pitch angle:

![Graph showing test set regression performance](image-url)
Vision-state fusion: Results

Test set regression performance, $R^2$ score (the higher the better) vs. pitch angle:

Outcomes:

1. Synthetic Pitch Augmentation has a strong effect across the entire pitch range (regularization effect).
Vision-state fusion: Results

Test set regression performance, $R^2$ score (the higher the better) vs. pitch angle:

Outcomes:
1. *Synthetic Pitch Augmentation* has a **strong effect** across the entire pitch range (regularization effect).
2. *Vision-state fusion* has stronger effect on $z$. 

---

D. Palossi
Vision-state fusion: Results

Test set regression performance, $R^2$ score (the higher the better) vs. pitch angle:

Outcomes:

1. *Synthetic Pitch Augmentation* has a **strong effect** across the entire pitch range (regularization effect).
2. *Vision-state fusion* has stronger effect on $z$.
3. The two techniques combined achieve the top performance.
Vision-state fusion: In-field evaluation

Closed loop evaluation of whole-system control performance, execution entirely onboard.

**Task:** autonomously approach the human subject, 1.3m in front at eye-level.

**Test setup:** subject stands still for 6s, then kneels → stresses model output variables $x$ and $z$. 
Vision-state fusion: In-field evaluation

Test setup:
5 test flights each for the stateless and stateful models (both with synthetic pitch augmentation), repeated over three subjects never seen in training. Total: 30 test flights.
Test setup:
5 test flights each for the *stateless* and *stateful* models (both with synthetic pitch augmentation), repeated over three subjects never seen in training. Total: 30 test flights.

Results:
*Stateless* struggles to track the subject, both horizontally and vertically.
Vision-state fusion: In-field evaluation

Test setup:
5 test flights each for the stateless and stateful models (both with synthetic pitch augmentation), repeated over three subjects never seen in training. Total: 30 test flights.

Results:
Stateless struggles to track the subject, both horizontally and vertically. Stateful follows the moving subject and firmly converges.
Conclusion

The fun has just begun…
● Self-supervised learning
● Novel ULP sensors
● On-device training
● Neural Architecture Search
● and more

Many thanks to all Ph.D. and Master students working on these topics from IDSIA USI-SUPSI, University of Bologna, and ETH Zürich.

Our PULP Team recently won the 1st Nanocopter AI Challenge @ IMAV’22 (TU Delft, NL)
Questions?
Backup
Example application: Autonomous navigation

Incremental modifications of the Dronet [4]:

- Batch Normalization Folding
  - Direct and inverse folding due to the by-pass topology
- Fixed-point 16 bit instead of Float 32 bit
  - Signed Q5.11 directly in training
- Max pooling 2×2 instead of 3×3
  - Minimal loss in accuracy with 2.25× reduction time
- Training dataset extension
  - ~1300 collision images with the same onboard camera

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Max pooling</th>
<th>Data Type</th>
<th>Regression (steering)</th>
<th>Classification (collision)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>EVA</td>
<td>RMSE</td>
</tr>
<tr>
<td>Original</td>
<td>3 × 3</td>
<td>Float32</td>
<td>0.758</td>
<td>0.109</td>
</tr>
<tr>
<td></td>
<td>3 × 3</td>
<td>Fixed16</td>
<td>0.746</td>
<td>0.115</td>
</tr>
<tr>
<td></td>
<td>2 × 2</td>
<td>Float32</td>
<td>0.766</td>
<td>0.105</td>
</tr>
<tr>
<td></td>
<td>2 × 2</td>
<td>Fixed16</td>
<td>0.795</td>
<td>0.097</td>
</tr>
<tr>
<td>Extended</td>
<td>3 × 3</td>
<td>Float32</td>
<td>0.764</td>
<td>0.104</td>
</tr>
<tr>
<td></td>
<td>3 × 3</td>
<td>Fixed16</td>
<td>0.762</td>
<td>0.109</td>
</tr>
<tr>
<td></td>
<td>2 × 2</td>
<td>Float32</td>
<td>0.747</td>
<td>0.109</td>
</tr>
<tr>
<td></td>
<td>2 × 2</td>
<td>Fixed16</td>
<td>0.732</td>
<td>0.110</td>
</tr>
</tbody>
</table>

Testing set evaluation on TensorFlow/Keras

Original

Proposed
Example application: Autonomous navigation

Maximize the parallel computation → high energy efficiency

- **Spatial** scheme (horizontal)
  - Requires sufficient amount of parallel work within the same tile
  - Used for the 1st layer (conv + max pooling)

- **Feature-wise** scheme
  - Heavier memory footprint + reduction stage
  - Used for all the remaining layers

```python
# weight DMA-in
# tiling over H
for t in range(nb_tiles_H):
    # prologue operation
    # tiling over K_in
    for j in range(nb_tiles_Kin):
        # input tile DMA-in
        for i in range(K_out):
            # body operation
            ...  
        # output tile DMA-out
```

```python
# tiling over K_out
for i in range(nb_tiles_Kout):
    # weight DMA-in
    # prologue operation
    # tiling over K_in
    for j in range(nb_tiles_Kin):
        # input tile DMA-in
        # body operation
        # epilogue operation
    # output tile DMA-out
```
Example application: Autonomous navigation

**DroNet**

Direct folding

\[ BN(W \ast x + b) = W' \ast x + b' \]

\[ W' = \sigma/\gamma \cdot W \]

\[ b' = b + \sum \mu \cdot \sum \alpha W' \cdot \sum (\beta + \sigma/\gamma \cdot \sum \beta W') \]

**BN folding reduces memory footprint and computation**

Direct folding

- Linear transformation of \( W \) and \( b \)

Inverse folding

- Result

Result
PULP-Dronet: In-field evaluation

Longest indoor traveled distance → ~113m

Collision avoidance capability
- Straight flight @ 1.5 m/s
- Dynamic obstacle appears at 2 meters
- Only 2 meters to brake

Floor plan

Distance

Altitude

2 m

0.5 m

8 m

10 m

Speed 1.5 m/s

Obstacle

YouTube @ PULP Platform Channel
**PULP-Dronet v2**

<table>
<thead>
<tr>
<th>Training</th>
<th>Testing</th>
<th>Original dataset</th>
<th>Himax dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original + Himax</td>
<td>RMSE, Accuracy, Accuracy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PULP-Dronet v1</td>
<td>float32</td>
<td>0.109</td>
<td>0.964, 0.900</td>
</tr>
<tr>
<td></td>
<td>int16</td>
<td>0.110</td>
<td>0.977, 0.891</td>
</tr>
<tr>
<td>GAPflow</td>
<td>float32</td>
<td>0.118</td>
<td>0.893, 0.905</td>
</tr>
<tr>
<td></td>
<td>int8</td>
<td><strong>0.120</strong></td>
<td>0.892, <strong>0.900</strong></td>
</tr>
<tr>
<td>NEMO/DORY</td>
<td>float32</td>
<td>0.136</td>
<td>0.925, 0.881</td>
</tr>
<tr>
<td></td>
<td>int8</td>
<td>0.135</td>
<td><strong>0.925</strong>, 0.886</td>
</tr>
</tbody>
</table>

Regression/Classification performances comparable with PULP-Dronet v1 despite the stronger 8bit quantization.
PULP-Dronet v3

![Graph showing BCE and MSE for Baseline Model and Tiny Model](image)
Dataset augmentation: Synthetic pitch

Task-specific data augmentation to generate multiple synthetic views at a different pitch from a single training image.
We explore 4 architecture variants to feed the state information to the model.
About me

I’m Daniele Palossi, Postdoctoral researcher at the Dalle Molle Institute for Artificial Intelligence (IDSIA) in Lugano and ETH Zürich.

daniele.palossi@idsia.ch
PULP Silicon Prototypes

History of the PULP:

Copyright 2021 © ETH Zürich
http://asic.ethz.ch/applications/Pulp.html

Credit: Daniele Palossi
Copyright Notice

This presentation in this publication was presented at the tinyML® EMEA Innovation Forum 2022. The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.