CODAI 2023 Workshop

Temporal Patience

Efficient Adaptive Deep Learning for Embedded Radar Data Processing

Max Sponner
Development Center
Infineon Technologies Dresden GmbH & Co. KG
2023-09-21

Co-Authors: Julius Ott, Lorenzo Servadei, Bernd Waschneck, Robert Wille, Akash Kumar

Table of contents

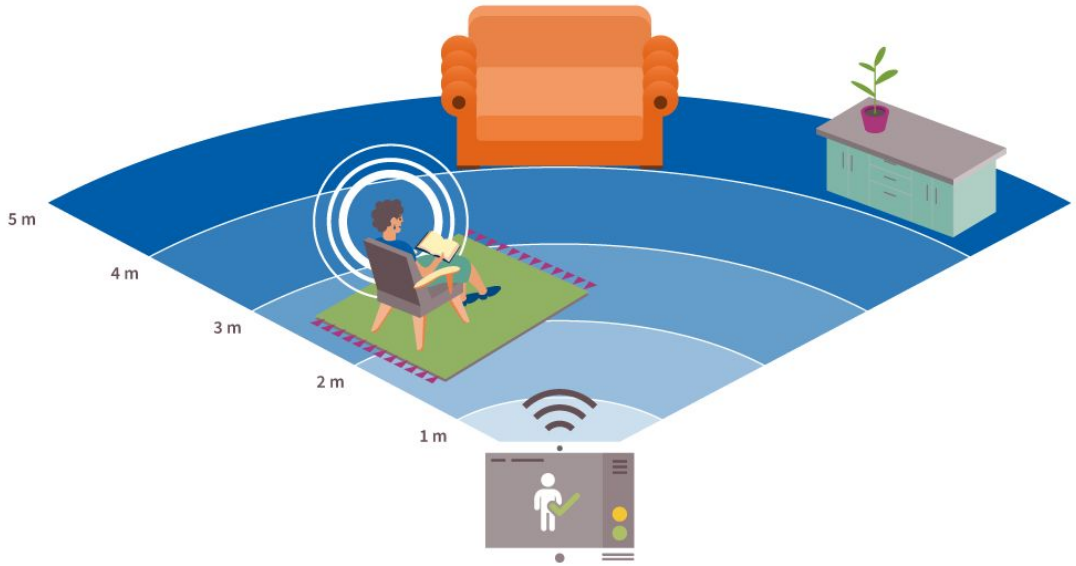
Table of contents

Radar-based People Counting

- Estimates the number of occupants in a given area

- all-in-one solution
  - “smart” sensor
  - Intelligent solution
  - Only outputs estimated number

- No…
  - Cloud infrastructure
  - Edge server
  - Network connection
  - Need for any additional compute



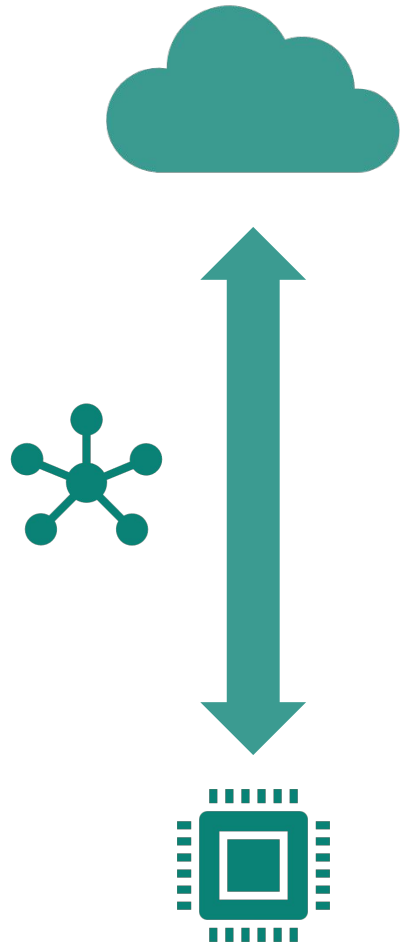
Benefits of local Processing

Privacy
- Data stays on the smart sensor
- Not send to any (third) party

Flexibility
- Can be deployed everywhere
- Easy to integrate into existing products

Independence
- No reliance on cloud
- No additional compute required
- No upkeep cost after sale
- No end-of-life due to API changes

Power Efficient
- Enables “virtual” always-on devices
- Powering down the main device
- Only smart sensor stays on



Problem: Performance

- Deployment has become easier
  - Modern toolchains automate most steps
  - Just need to find appropriate network architecture

- The solution works, but…
  - Rather slow
  - High power consumption

- Can we improve this further?
  - Further optimizations reduce accuracy too much

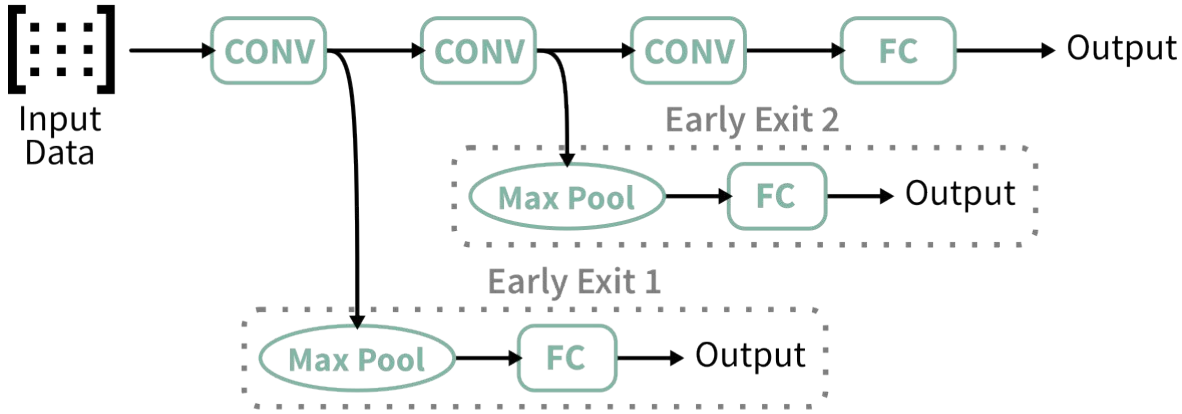
Table of contents

Adaptive Neural Networks

- Traditional Deep Learning: static
  - Neural Networks: no complex control flow
    - Easy to optimize/pre-plan execution
    - Limits its capabilities; every sample is treated the same
  - Optimizations: permanently impair prediction capabilities
    - Applied ahead-of-time
    - i.e. pruning, quantization, …
- Observation:
  - Not all samples are equal
  - “easy” and “hard” samples exist
- Adaptive Neural Networks:
  - Change structure and parameters at runtime
  - Depending on current circumstances (input, resources, …)

Early Exit Neural Networks

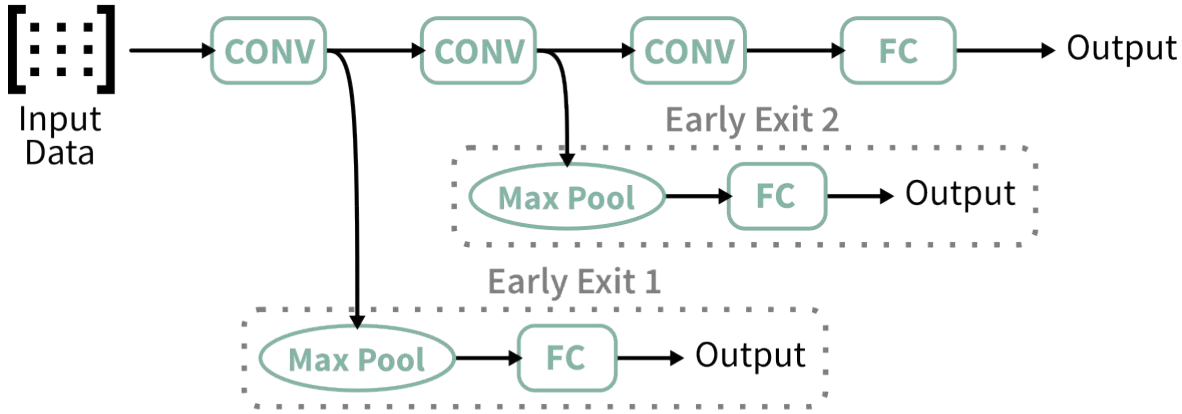
- Can terminate inference early
- Introduce Early Exits:
  - Additional classifiers
  - Between hidden layers
  - Trained for the same task as the final classifier
- Inference can be terminated if early result is sufficient
- Idea: Spend only as many resources as necessary
  - “Simple” data terminates early
  - “Hard” samples classified correctly by full network
- Can be seen as dynamic layer pruning



Early Exit Neural Networks

- Problem: how to decide when it is “sufficient”?
  - Small footprint, fast computation
  - Good decisions

- State-of-the-art
  - Rule-based:
    - Budget
    - Confidence
    - Patience
  - Agent-based:
    - Additional model
    - Trained to select classifier
    - Not suitable for embedded



Observations

- Radar sensor data is temporally correlated

- Subsequent samples are often similar
  - Time frame depends on the application
  - Conference room: hours/days without occupants

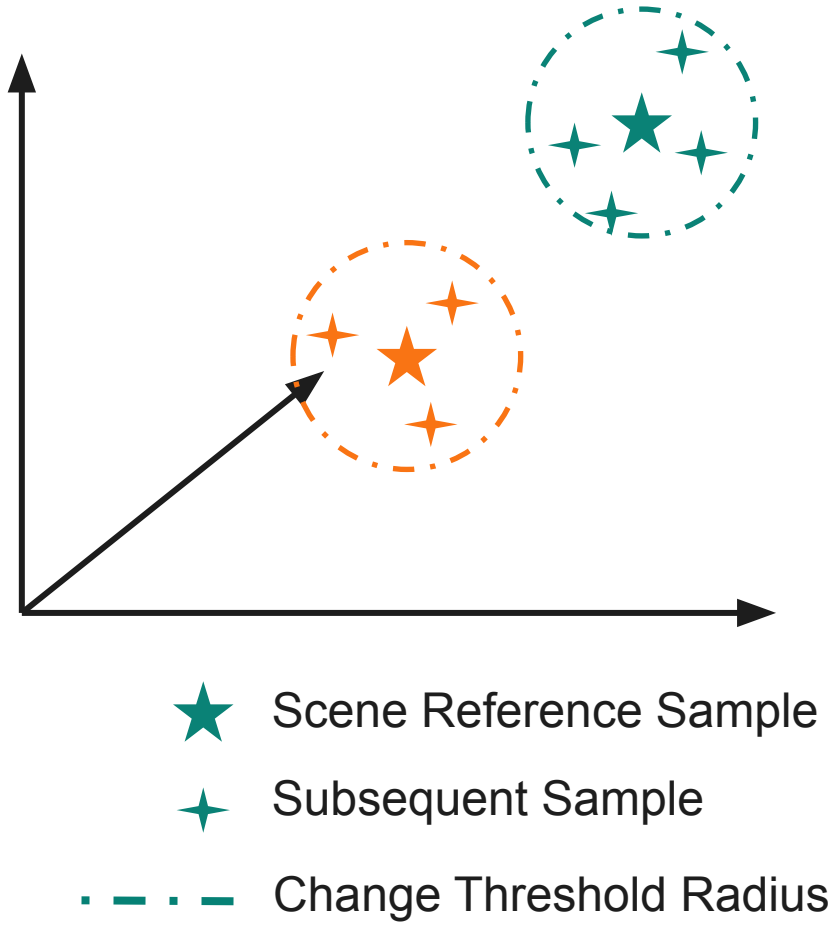
- Unnecessary to perform the entire inference again
- Just need to detect if it is similar
  - No reliable similarity metrics for radar data
  - Not expressive enough
  - Too expensive to run on every sample

- Idea: Monitor early results instead
  - Change in output should correlate to change in input
  - Utilizes extracted features that are relevant for task

Table of contents

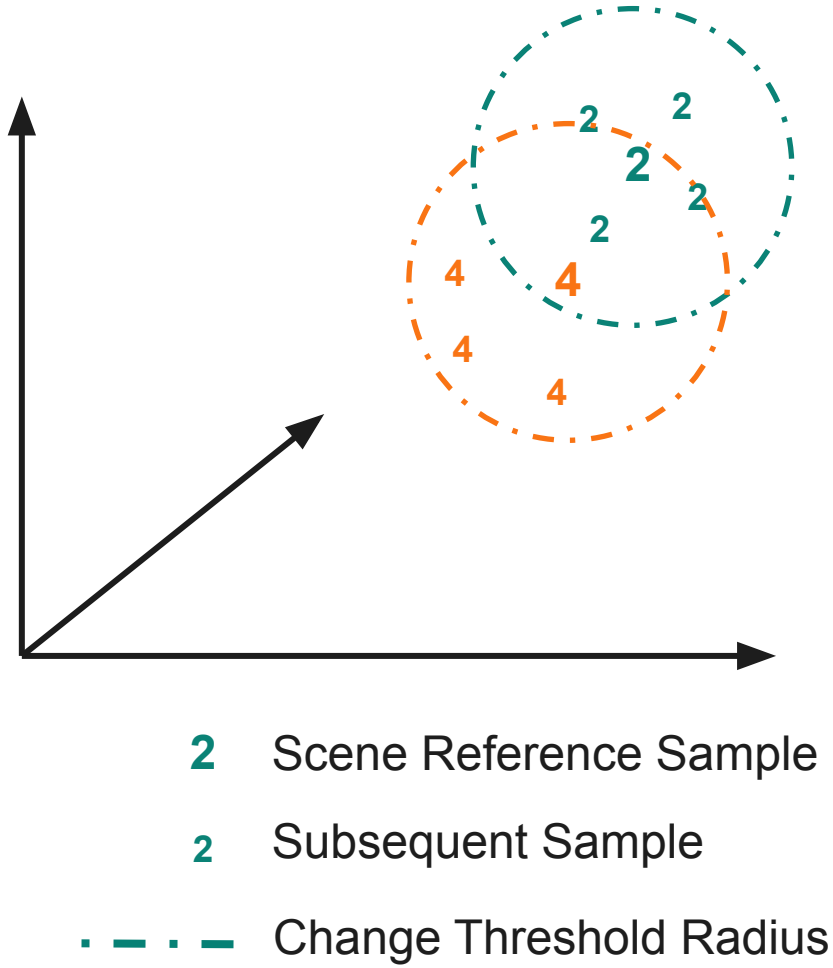
Difference Detection

- Observe change in first Early Exit
- Calculate distance, compare to threshold
  - If below threshold: terminate, reuse previous result
  - Otherwise: continue inference
- What to use as the reference:
  - Direct predecessor
  - would lead to accumulating small changes
  - “Scene Detection” instead
- “Scene”
  - Group of subsequent similar inputs
  - Always use the first sample as a reference
  - Only label the first sample by full inference
- New Scene:
  - Started by a sample that is different from previous



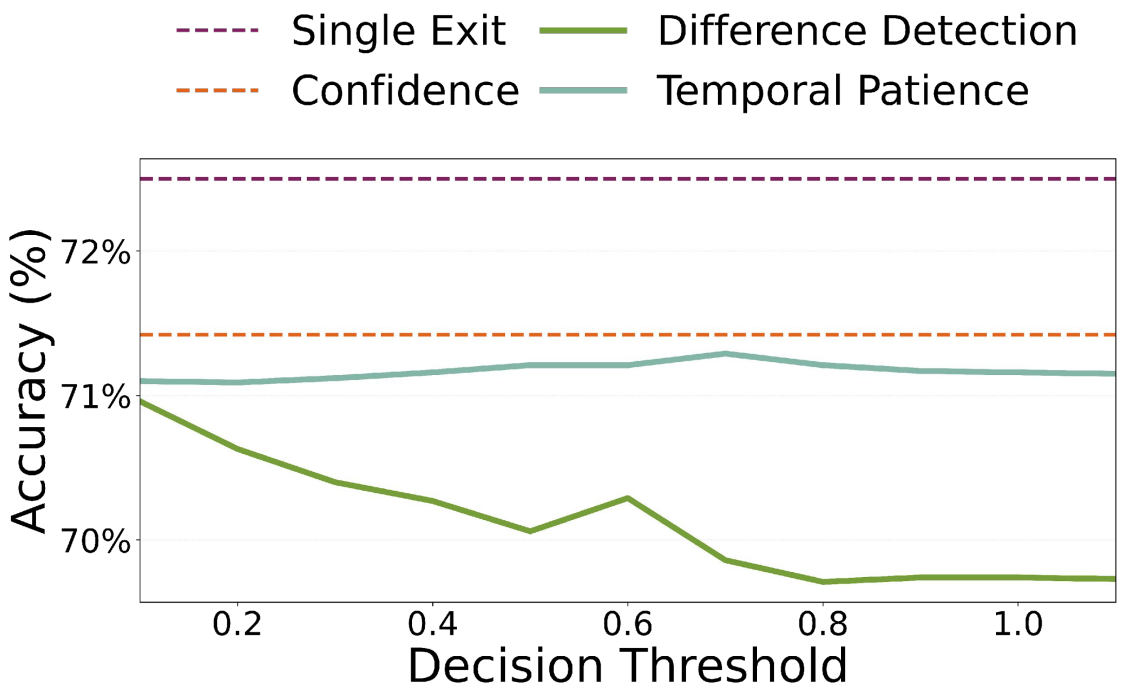
Temporal Patience

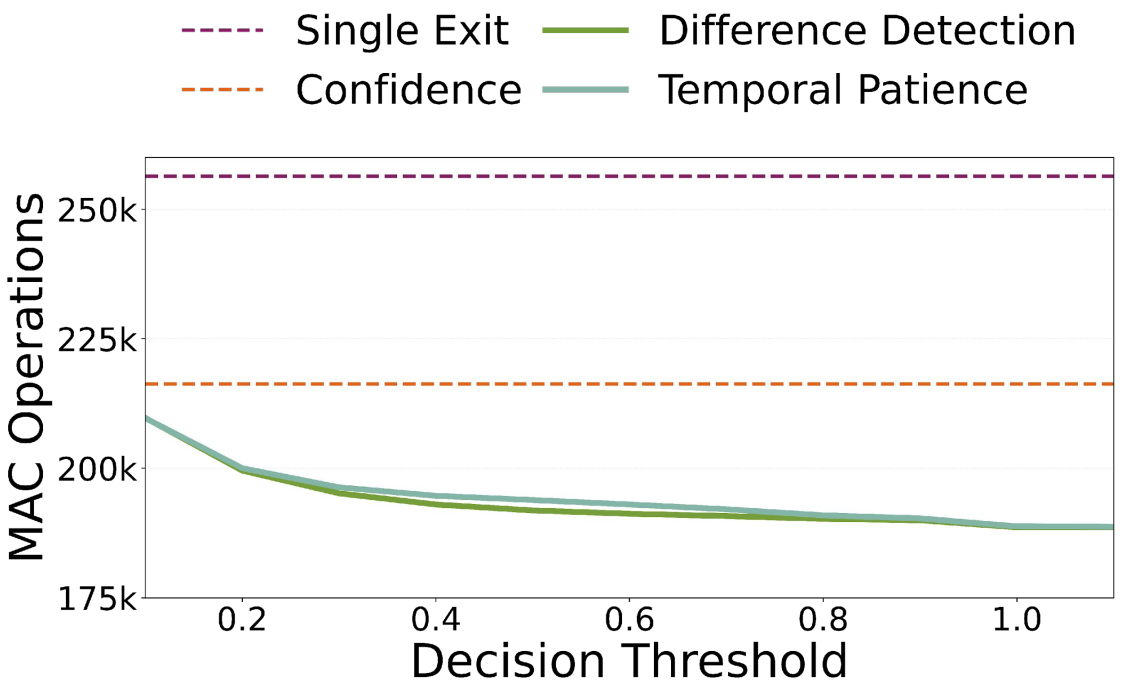
- Select used classifier dynamically
  - Higher computational cost
  - But increased accuracy
- Idea: use classifier that can detect relevant features
- New Scene Detection:
  - Not only reliant on distance
  - Also considers output label
  - Increases accuracy in ambiguous situations



Benchmark

- Evaluated the performance on radar recordings
- Duration: 30 seconds – 2 Minutes

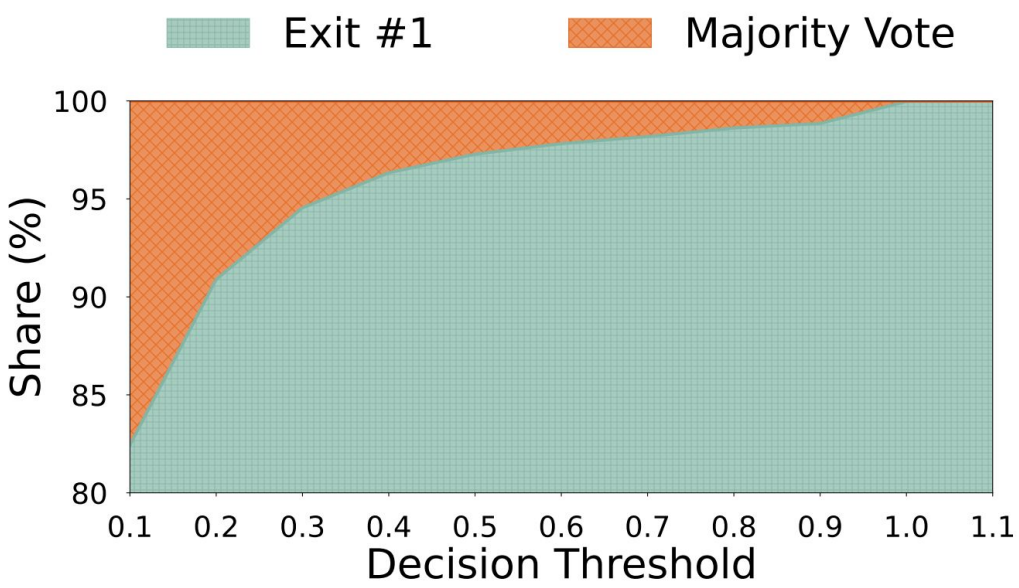




Benchmark

- Evaluated the performance on radar recordings
- Duration: 30 seconds – 2 Minutes

Difference Detection



Temporal Patience

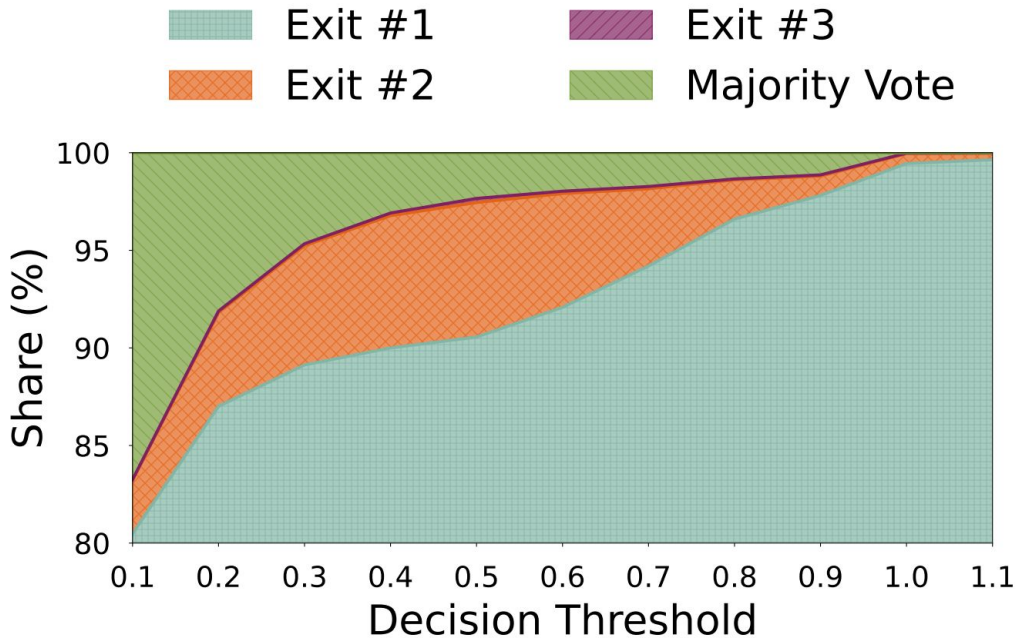


Table of contents

Conclusion

- 26% less mean operations per inference

- Very early results
  - On a single application
  - Short recordings, even shorter Scenes
  - Applications with more static content could increase savings

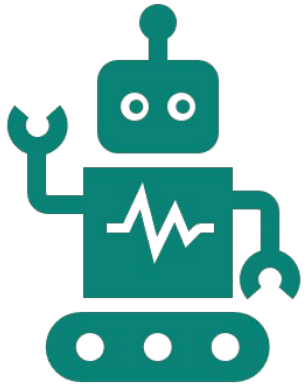
- Temporal Component:
  - RNN: leverage for prediction quality
  - Temporal Patience: leverage for better efficiency

  - Future research on utilizing it for increased efficiency

Future Work

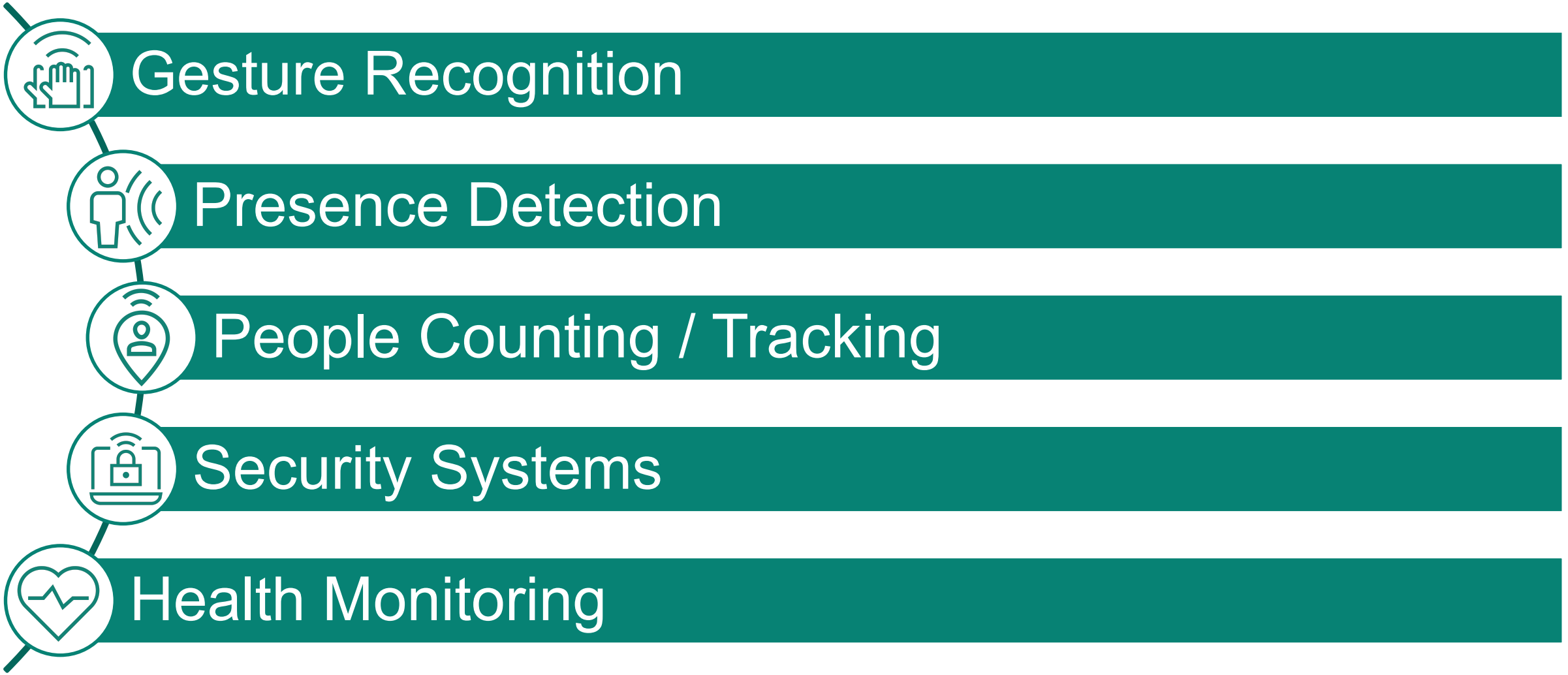
- Different Data Modalities
  - Not only radar data is temporally correlated
  - IoT also utilizes various sensors
    - Microphones
    - Cameras
    - Specialized sensors

- Democratize Adaptive Deep Learning
  - Niche topic, not well known
  - Requires expertise
  - Enable automatic deployment
    - Search for a suitable Adaptive solution
    - Network adaptation method
    - At-runtime decision mechanism
    - Retraining of rewritten adaptive model



The project on which this report is based was funded by the German Ministry of Education and Research (BMBF) under the project number 16ME0542K. The responsibility for the content of this publication lies with the author.

Applications for Radar Sensors in IoT



Applications for People Counting

Smart Home / Building

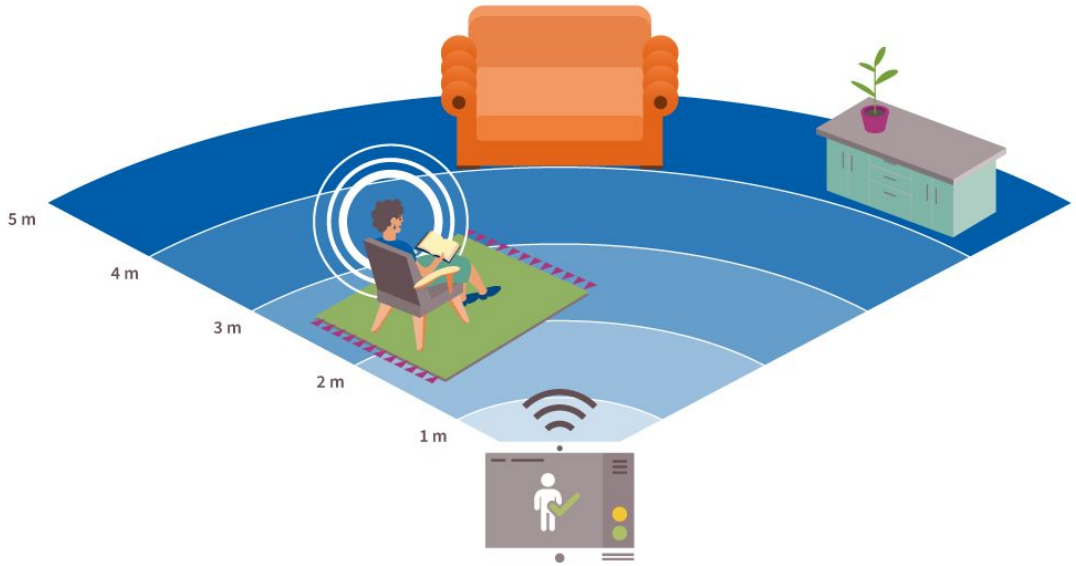
– Control building based on utilization

“Virtual” Always-On Devices

– Automatic wake-up
– Only active when users are present

Security Systems

– Alarm system
– Personal device privacy detection



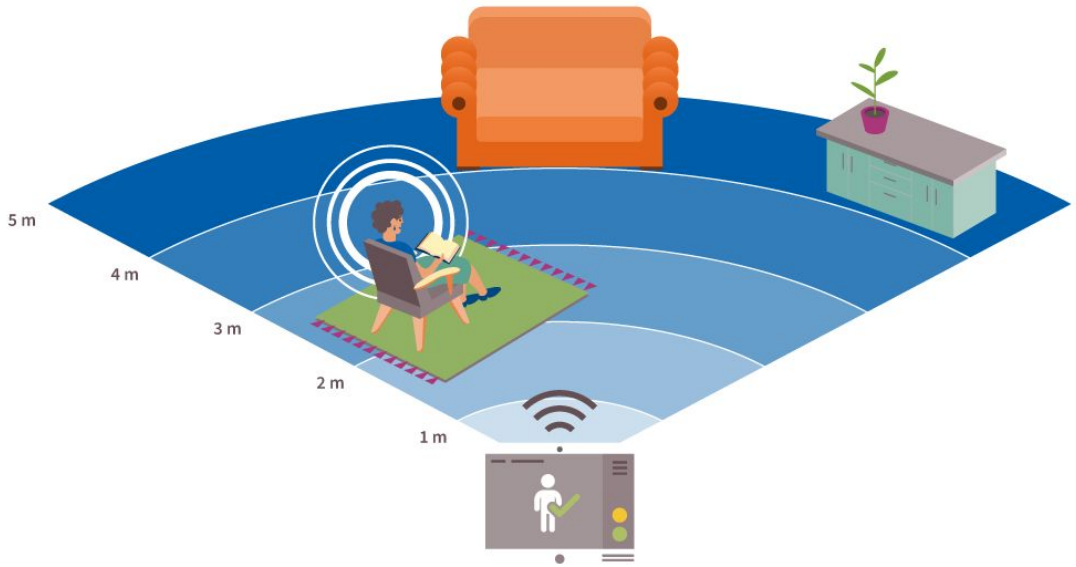
Application: Radar-based People Counting

- Task: Estimate the number of people
  - In a given area
  - Using radar sensor data

- Solution:
  - 60 GHz Radar Sensor
  - Cortex-M Microcontroller
  - Neural Network

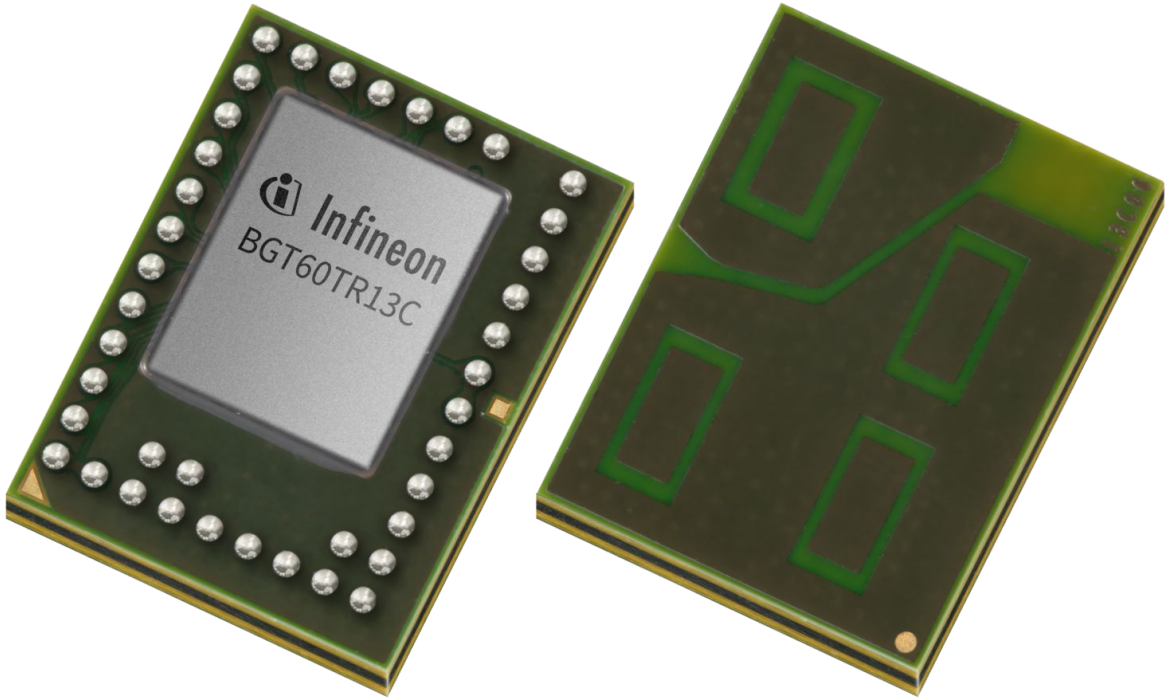
- Use case:
  - Home Automation
  - Always-on Devices
  - Security Features

- Problem:
  - Fitting the Workload onto the MCU
  - Preprocessing + Inference + Postprocessing
  - Minimize latency and power consumption

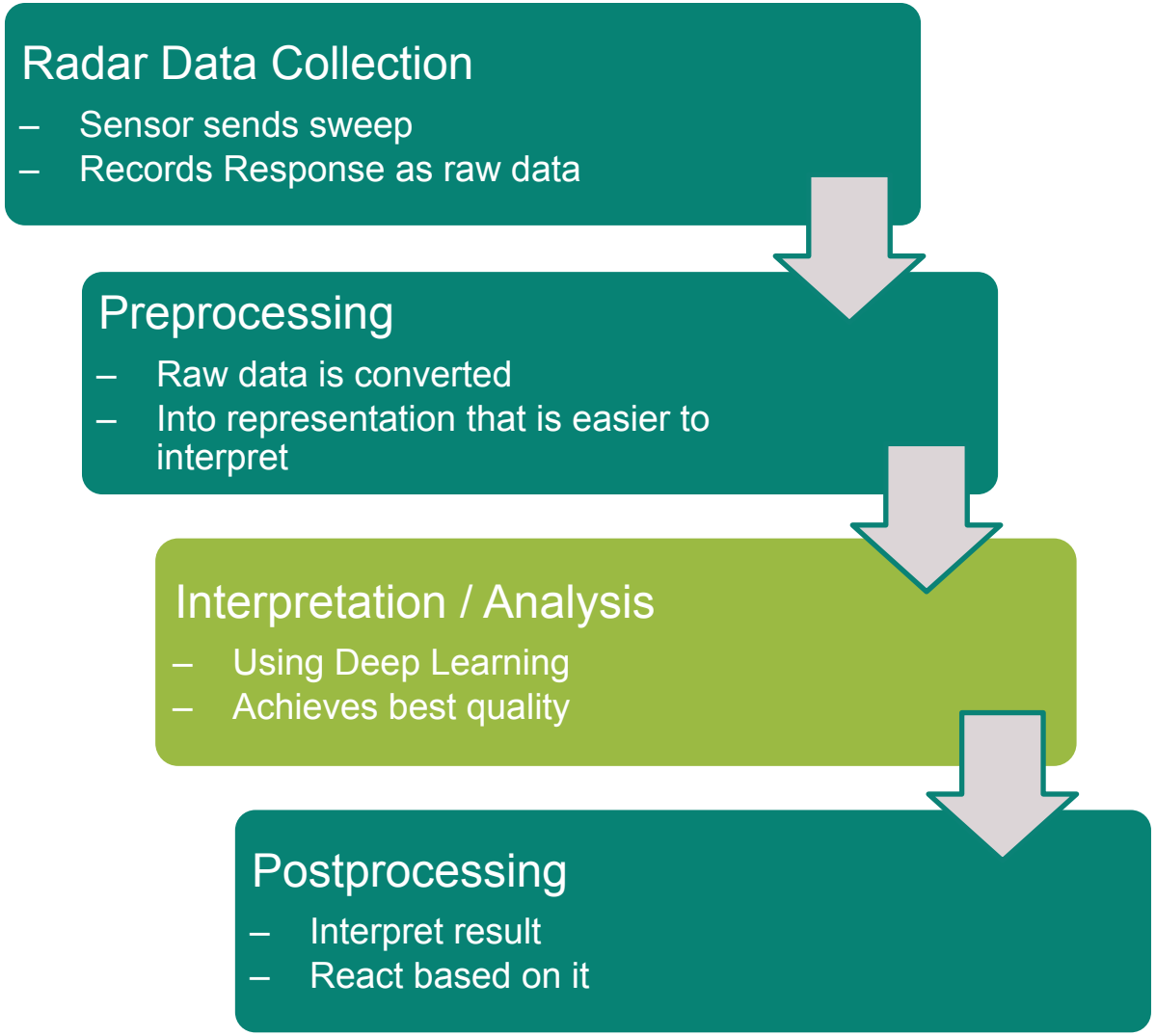


Why Radar Sensors?

Small footprint	– Low Power Consumption – Small Size
Environment independent	– Does not need Lighting Source – Robust against Weather Conditions – Not harmful for Pets
Privacy Preserving	– No Biometric Features – Not Recognition of Person
High Quality Data	– High Resolution – Medium Range – Reconfigurable



Processing Steps



Problem: Processing the Data (locally)

- Radar data must be analyzed to perform the task
  - Preprocessing + Deep Learning Inference
  - Deep Learning enables the best accuracy

- Processing should stay local
  - Maintain independence
  - Maximal flexibility
  - Not all use cases can rely on the presence of more compute resources

- But: “local” = Cortex-M class of devices
  - Needs to perform preprocessing + inference
  - Processing latency needs to be short
  - Power consumption should be as low as possible
  - MCU is limited in compute power

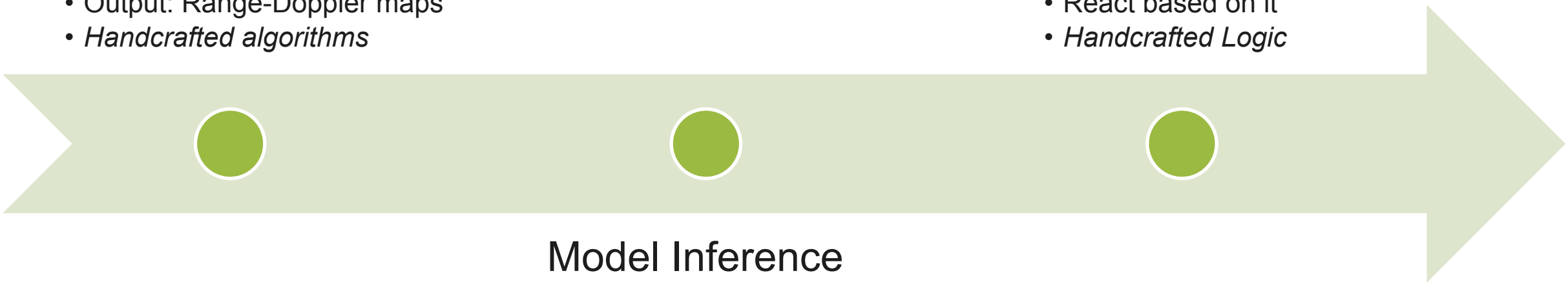
Data Processing

Preprocessing

- Input: raw sensor data
- Output: Range-Doppler maps
- *Handcrafted algorithms*

Postprocessing

- Interpret prediction result
- React based on it
- *Handcrafted Logic*



Model Inference

- Input: Range-Doppler Maps
- Output: Prediction
- *Convolutional Network*

Problem: Processing the Data (locally)

- **One package: Sensor + MCU + Data Processing**
- MCU
  - Cortex-M class device
  - Perform the entire data processing pipeline

Data Processing

- Preprocessing
- Deep Learning Inference
- Postprocessing

- Deep Learning achieves best quality results

Local Execution

- Maximal flexibility
- Maintain independence
- No reliance on additional compute resources

- Allows for coverage of all use cases

Constraints & Requirements

- Users demand short latency
- Low power consumption
- Accurate results

- MCU:
  - Limited compute power
  - Not always a network connection