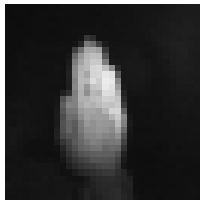
Gesture recognition on RH850

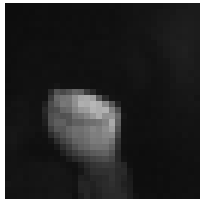
Problem characterization

Given an image from a sensor, select the right category. Computation to be performed at RH850.

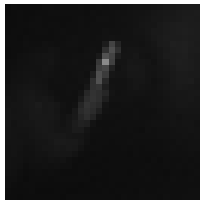


PALM

OPEN



FIST



OTHER/NONE

Gesture recognition options

- Manual characterization of selected gestures and manual feature detection
  - For instance:
    - FIST … oval object without a hole inside

    - PALM … more pointed oval object than fist with convex hull similar to the object's mask

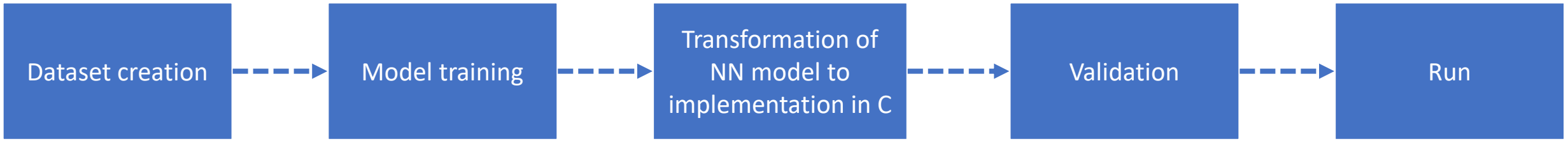
    - OPEN … object with a convex hull significantly having significantly higher density than object's mask

- Using a neural network
  - No manual characterization of features is needed – the neural network learns these characteristics itself from training data

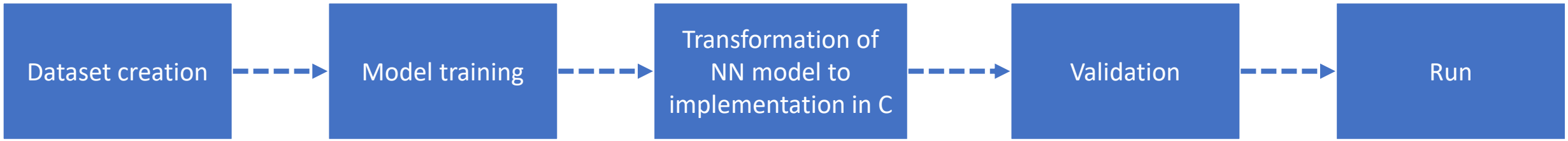
Neural network considerations

- No need to manually characterize gestures
  - Reduces the developer's bias
  - Makes it easier to introduce new gestures
- Training requires a good amount of data
  - Typically, at least thousands of samples
- The achievable accuracy depends on the amount and variety of training data
- No easy way to understand how the neural network characterizes the particular gestures
  - It works as a black-box
- Validation has to be done through success on a validation dataset
- There exist mature frameworks for neural network training
- … and reference architectures for neural network – e.g. for character recognition (e.g. LeNet-5)

Gesture recognition process and toolchain



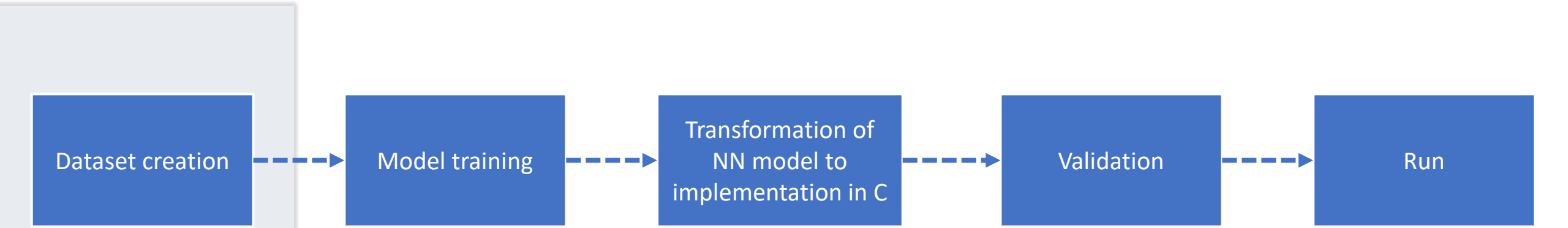
Gesture recognition process and toolchain



Challenges and specifics:

- Very limited memory and computation resources to execute the neural network (RH850 target platform)
- Providing measurable reliability and a suitable form of guarantees

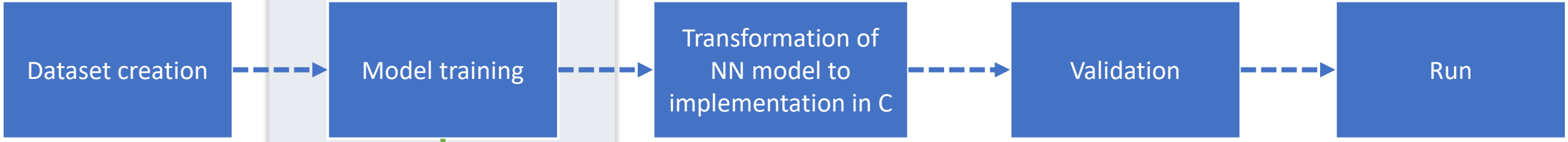
Gesture recognition process and toolchain

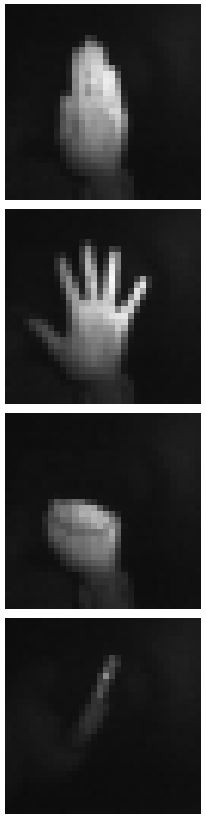


Datasets

- Experimental first version of a datasets
- 4 categories: PALM, OPEN, FIST, OTHER/NONE
- Training dataset – approx. 1500 images
- Validation dataset – approx. 300 images

Gesture recognition process and toolchain



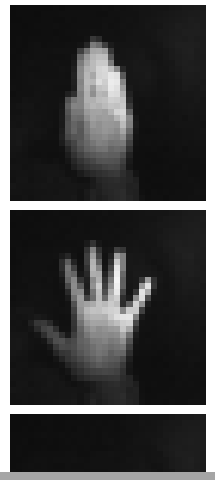


NN model based on reduced LeNet-5 architecture

Training using Tensorflow framework
Implemented in Python

Layer		Feature map	Size	Kernel size	Stride	Activation
Input	Image	1	32x32	-	-	-
1	Convolution	5	30x30	3x3	1	Relu
2	Average pooling	5	6x6	5x5	5	Relu
3	Convolution	9	4x4	3x3	1	Relu
4	Average pooling	9	2x2	2x2	2	Relu
5	Dense	-	32	-	-	Relu
6	Dense	-	16	-	-	Relu
Output	Dense	-	4	-	-	Softmax

Gesture recognition process and toolchain



Dataset creation

Model training

NN model based on reduced LeNet-5 architecture

Transformation of NN model to implementation in C

Validation

Validation executor

Validation data for gesture detector

Run

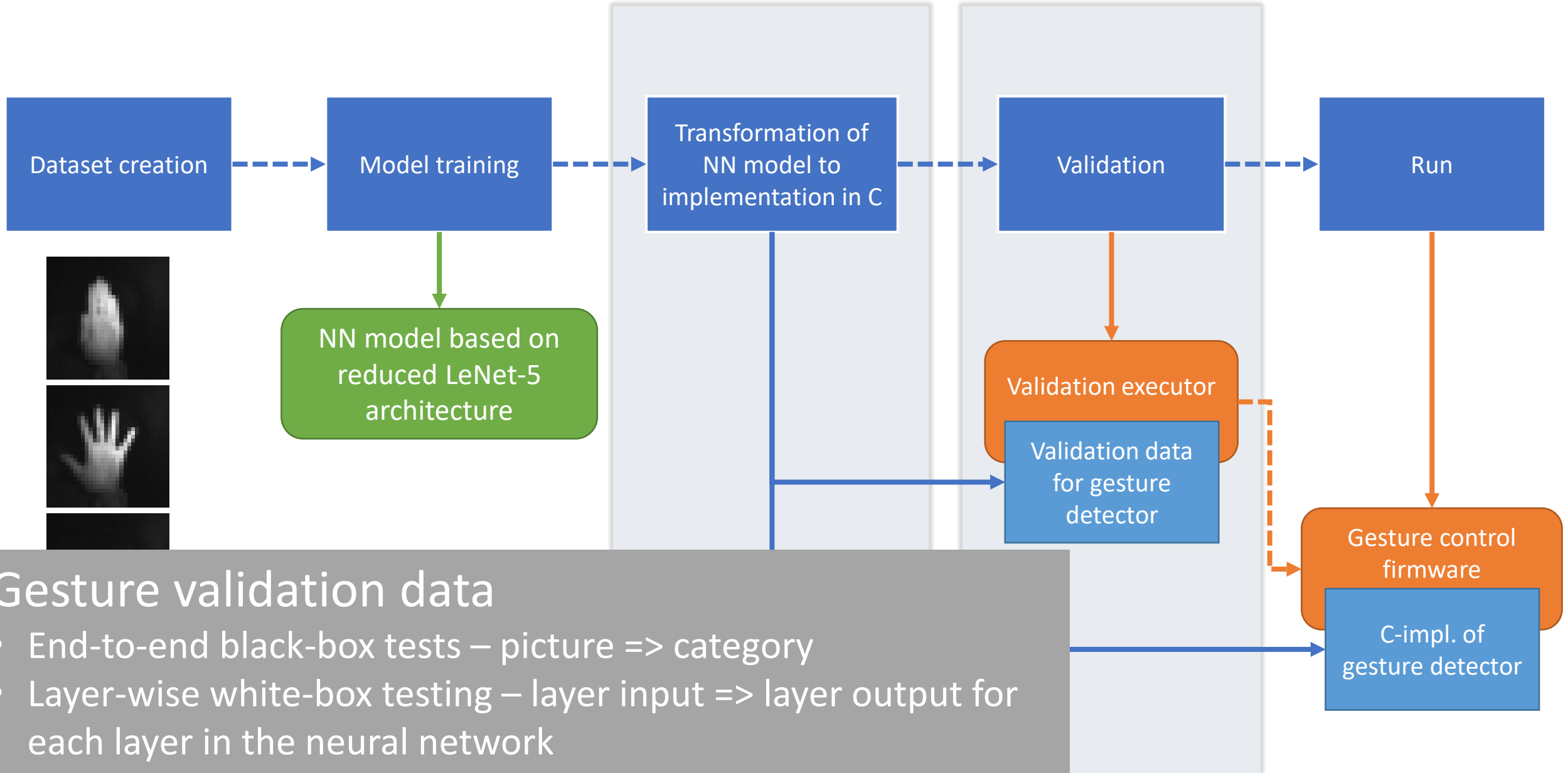
Gesture control firmware

C-impl. of gesture detector

Gesture detector implementation

- Low-footprint implementation in plain C language
- No dependencies except for math.h and float.h
- Can be directly used with RH850 compiler

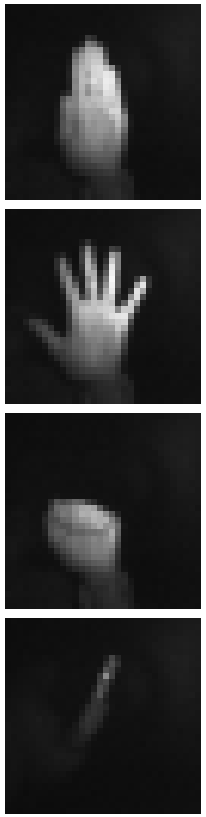
Gesture recognition process and toolchain



Gesture validation data

- End-to-end black-box tests – picture => category
- Layer-wise white-box testing – layer input => layer output for each layer in the neural network

Gesture recognition process and toolchain



Dataset creation

Model training

Transformation of NN model to implementation in C

Validation

Run

NN model based on reduced LeNet-5 architecture

Validation executor

Validation data for gesture detector

Gesture control firmware

C-impl. of gesture detector