# Complexity bounded classification of fish-eye distorted objects with micro-controllers

Danilo Pau, Alessandro Carra
STMicroelectronics

Marco Garzola
Tecniplast

## Introduction

Measuring individual rats or mice's activity in their home cage provides useful information such as the welfare of the laboratory's animals, the monitoring of their daily activities like drinking, eating. These studies also help basic research on drug discovery and associated developments.

In this context, we developed three complexity bounded Convolutional Neural Network (CNNs), named 1)µCageNet, 2)µBottleNet and 3)µFoodNet. They can classify the presence, or the absence respectively of the cage and the water bottle, while the third one classifies the level of the food into the feeder. These CNNs achieved 99% accuracy on an in-house made dataset created using ad-hoc hardware boards for image acquisition. Since at our best knowledge, no public datasets were available to support this study, we built up to five versions of three image datasets to train the three neural networks.

From image capture to the inference's execution, the tasks have been executed by either STM32L4 (ultra-low-power core that consumes as low as 120 µA/MHz) and the STM32H7 micro controller units (MCUs). The CNNs have been designed to fit into the constrained MCU resources. Special attention has been given to the on-chip memory occupation to ensure the RAM footprint was as low as 39.44 KBytes.

To evaluate and test CNNs live on the field, the performances of these models (and against to the MobileNetV2 one), hardware boards and a graphical user interface (GUI) has been developed to gather the validation and test results of inference runs on the MCUs.

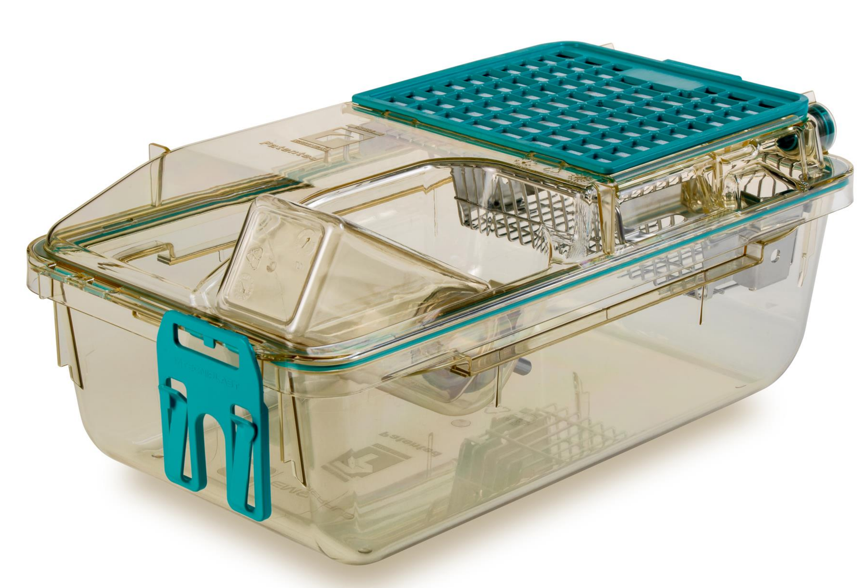


Figure 1 Exemplary rodent's cage with food feeder and water's bottle

## Claims

Authors desire to claim that no rats or any other animals were either involved, filmed or harmed in the making of the datasets. The image camera, used to capture pictures put inside the data set, did not capture any animals. And it's not specified to emit any kind of rays. It is based on a conventional passive image sensor and therefore not invasive to their skin or eyes.

## Dataset

The full resolution images (640x480 pixel) characterized by distortion introduced by fish eye lens, have been cropped as shown in figure 2.

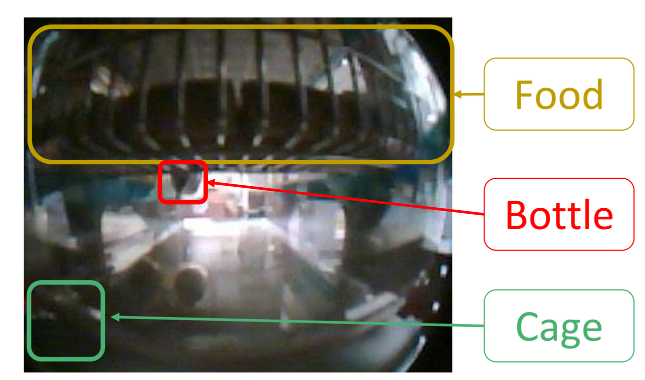
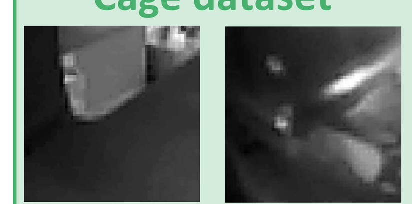



Figure 2 Exemplary full resolution image

**Cage dataset**



Cage-off Cage-on
Luma only
83,598 images
Res: 64x64
2 labels, balanced

**Food Dataset**

Food-empty Food-low Food-high
Luma only
68,430 images
Res: 102x200
3 labels, balanced

**Bottle Dataset**

Bottle-off Bottle-on
Luma only
30,901 images
Res: 50x50
2 labels, balanced

## Convolutional Neural Network developments

Three CNNs have been developed in Tensorflow/Keras and trained using floating point 32-bits precision with the datasets split in 80% train set (with 30% validation set) and 20% test set. Next, the pre-trained CNNs were quantized to integer 8 bits using Tensorflow lite post training quantization procedure. That step minimized model footprint by four times in term of ROM and RAM. That procedure proved to speed up execution compared to fp32 execution due to the X-CUBE-AI optimized ANSI C code generated.

The CNNs were featuring similar topologies to facilitate retraining and including the development of a single model which can be inferred on the cage, bottle and food dataset images.

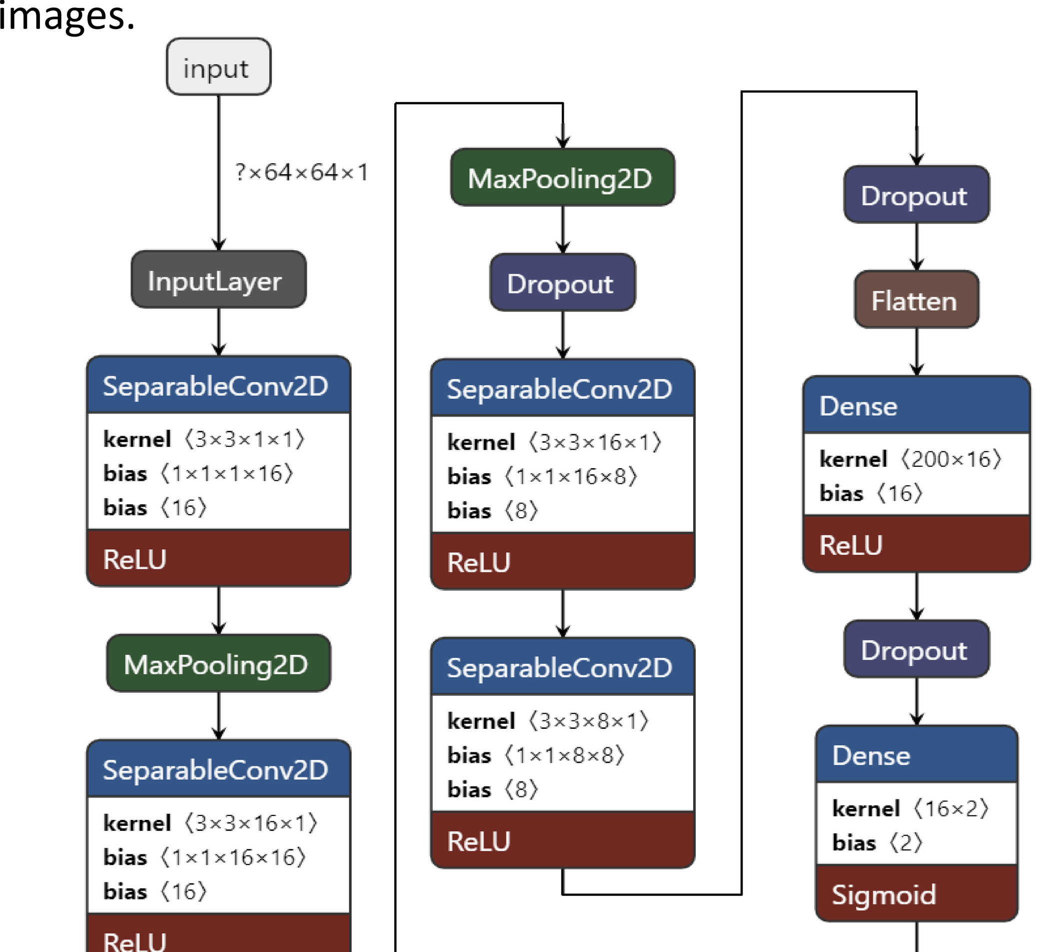



Figure 3 µCageNet topology

## Results and implementation on MCUs

**µCageNet**
MACC: 145,257
ROM: 4.39KB
RAM: 6.16KB
Inference time: 2.94ms on STM32H7 @480MHz
K-Fold Cross validation K=5 Accuracy: 99.9%

Cage detected? No / Yes

**µBottleNet**
MACC: 76,410
ROM: 2.39KB
RAM: 3.8KB
Inference time: 1.72ms on STM32H7 @480MHz
K-Fold Cross validation K=5 Accuracy: 99.7%

**µFoodNet**
MACC: 813,173
ROM: 26.05KB
RAM: 24.96KB
Inference time: 14.35ms on STM32H7 @480MHz
K-Fold Cross validation K=5 Accuracy: 99.9%

The CNNs with bounded model size and complexity have been designed specifically to perform adequately on each data set, maximizing accuracy and minimizing model footprint so that they could be deployed, all together, into either the STM32L4 and STM32H7 MCUs. µBottleNet and µFoodNet have been executed only if the cage would have been detected by µCageNet as shown in the flow graph beside.

The implementation analysis on the MCUs was possible, without time consuming C code hand-crafting, using X-CUBE-AI v.7.0.0, a free of charge tool-chain, that provided fast and productive way to automatically analyze, generate and validate the CNNs on the MCU's. The above graph reports also the complexity in terms of Flash [KiBytes], RAM [KiBytes], Multiply-and-ACCumulate (MACC) operations, inference time [ms]. Inference times were measured by executing the three CNNs, on STM32H7 @480MHz and on STM32L4 @80MHz and added together as shown in table 1.

| Table 1 | STM32H7 @480MHz | STM32L4 @80MHz |
|---|---|---|
| Inference: [ms] | 19.02 | 181.64 |

## Comparison with MobilenetV2

K-fold cross validation has been performed and the resulting performances of the two CNNs were compared to the MobileNetV2. The three proposed complexity bounded CNNs exceeded the accuracy achieved (+17%) by this more complex (65 times more weights) CNN architecture.

## Field tests

To help to evaluate and test the performances of these CNN models on the field, a graphical user interface (GUI), has been developed. It could render the validation and test results of inference runs on the MCUs using also some files or live shots from the image sensor connected to the PC.

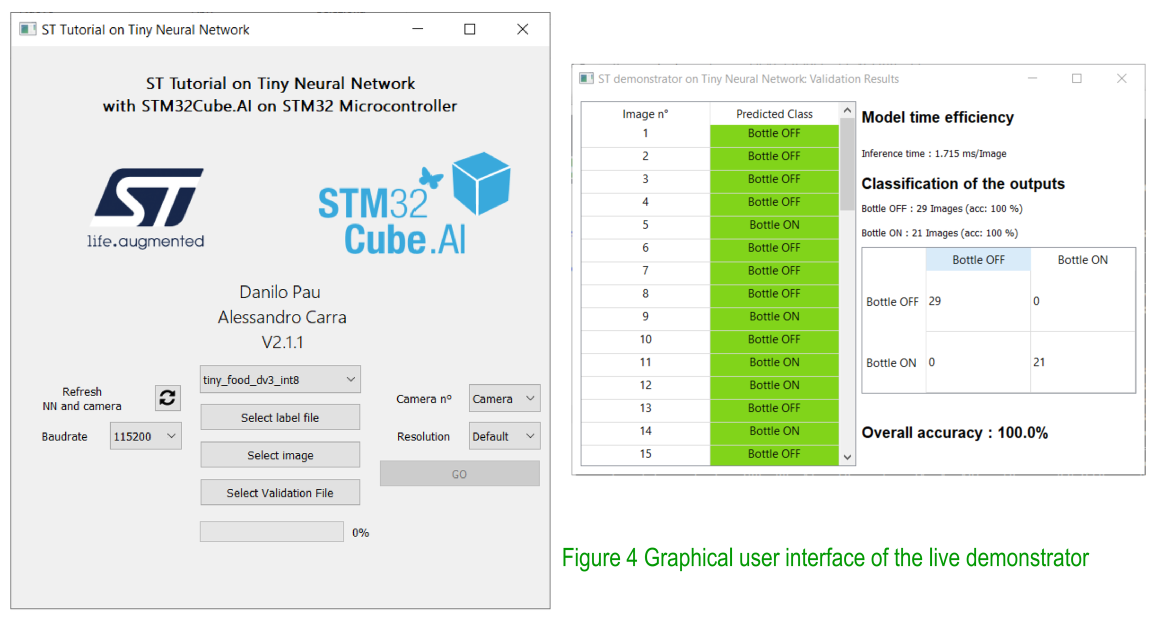



Figure 4 Graphical user interface of the live demonstrator

## Live deployment

The operator can keep under control all the cages (in a rack) and its main parameters thanks to the remote browser interface developed as shown in figure 5. On the left there are the live streaming of the cameras, one for each side. The data on the right are the results of the CNNs executed using the image squared in yellow.

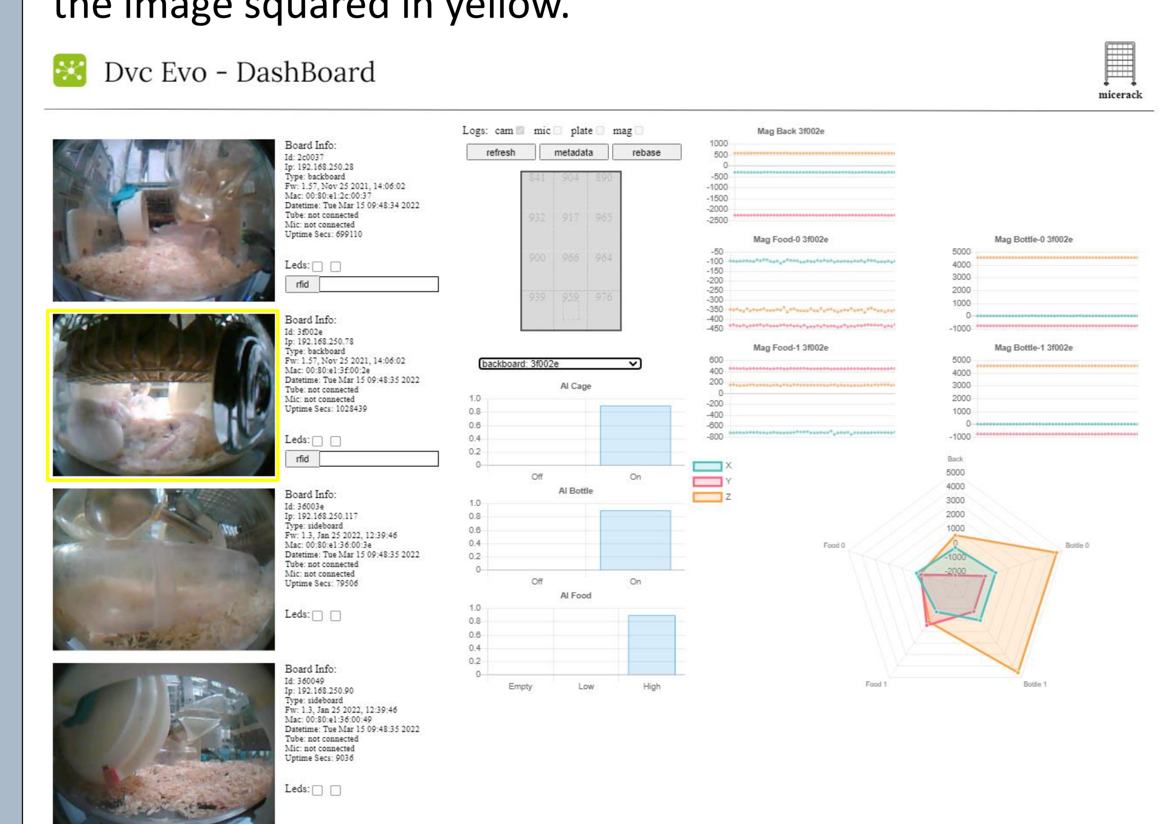



Figure 5 Remote user interface

## Conclusions

New Cage, Bottle and Food datasets have been created through multiple version from color to grayscale.

The µCageNet, µBottleNet and µFoodNet were developed to achieve the lowest RAM footprint, fit all together in the STM32 MCUs. Each one has achieved 99% accuracy score without any image pre-rectification using pre-trained models quantized to integer 8bits using TFLite post training quantization procedure.

Moreover, MobileNetV2 underperformed (-17% accuracy) on Food dataset with 65 times more parameters.