Smart Food Waste Management - Embedded Machine Learning vs Cloud Based Solutions

Raphael Zingg[1], Philipp Andermatt[2], Amin Mazloumian[1] and Matthias Rosenthal[1]

[1]Institute of Embedded Systems, Zurich University of Applied Sciences (ZHAW), Winterthur, Switzerland
[2]KITRO SA, Zürich, Switzerland

Abstract

In Switzerland, 2.8 million tons of food are lost or wasted across all stages of food production - every year. This equates to approximately 330 kg of food waste per person. By analysing and classifying discarded food with a smart waste analysis system combined with machine learning, valuable insights can be gained and the amount of wasted food can be significantly reduced. In this paper, we present how we have developed an embedded system which helps to solve this task. The embedded system operates in a decentralized manner: It captures an image every time food is thrown into a bin. The discarded food is identified and classified with machine learning algorithms. This provides a detailed insight into the structure of food waste for customers, e.g. restaurants or canteens. We implemented the machine learning algorithm directly on the embedded systems control unit. We found that running machine learning directly on embedded devices has many advantages compared to running them in the cloud: We saved significant amounts of cloud storage and reduced power consumption by up to a factor 100. In addition, privacy was increased and required bandwidth reduced because only the machine learning results are forwarded to the cloud, not the full data.

Keywords

Food Waste, IoT, Embedded Machine Learning, TensorRT, Embedded Inference

1. Introduction

In Switzerland, 2.8 million tons of edible food is wasted across all stages of the food industry. This equates to approximately 330kg of avoidable food waste per person and year, or 37% of agricultural production, i.e. food produced in Switzerland and abroad for consumption in Switzerland [1]. To produce all this wasted food, half of Switzerland's total farm land would be required. Moreover, this huge amount of wasted food is responsible for 27% of Switzerland's environmental impact [2]. The sectors retail, gastronomy and households contribute a total of 1.39 million tons of wasted food per year [2]. Most of this food waste is avoidable and comes from cooking surplus and consumer preferences (aesthetics) [3]. In addition to the environmental costs, it is estimated that the gastronomy sector in Switzerland loses 1 Billion Swiss Francs due to wasted food per year [2].

FTAL 2021, October 28–29, 2021, Lugano, Switzerland
zing@zhaw.ch (R. Zingg); philipp.andermatt@kitro.ch (P. Andermatt); mazl@zhaw.ch (A. Mazloumian); rosn@zhaw.ch (M. Rosenthal)
https://www.kitro.ch/ (P. Andermatt); https://blog.zhaw.ch/high-performance/ (M. Rosenthal)

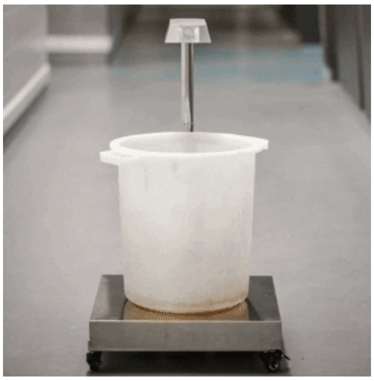


Figure 1: IoT device. The key hardware consists of a scale, a control unit and a camera.

To achieve the target number 12.3 of the Sustainable Development Goals (SDG) it is crucial to: "By 2030, halve per capita global food waste at the retail and consumer levels and reduce food losses along production and supply chains, including post-harvest losses" [4]. There exist different interventions which can lead to a reduction in food waste, such as size or type of plates or changing nutritional guidelines [5]. Interestingly, a recent study has found that by only providing food waste weight data to consumers and charging them accordingly, their food waste was reduced by 33% [6]. However, without detailed information about the wasted food it is difficult to make evidence-based decisions to prevent consummation stage food waste in a cost-effective manner [5]. Evidence-based decisions are only possible if there is information and data available about the subject. Regarding consumer based food waste it is crucial to have information about the kind and quantity of thrown away food items. Smart garbage systems, which collect waste data in a decentralized manner and transfer data to the cloud, can be used to collect this required data. Current IoT smart garbage systems are primarily used to ensure that no garbage bins are overloaded [7, 8]. However, there exist some smart garbage systems which are specialized in food waste [9, 10].

This paper shows how we have developed a smart IoT food waste monitoring system, shown in figure 1. It takes images and measures the weight of food waste in commercial kitchens and uses embedded machine learning (ML) to analyse the data. A summary of the collected data is provided to the customers via an online dashboard. The customers can track their waste over time and take data-driven decisions to reduce food waste.

Using machine learning directly on the embedded device offers many benefits and has a huge potential due to the omnipresence of embedded devices [11]. Moreover, running machine learning algorithms directly on the embedded system enables real-time data processing, increases privacy because only results and not the raw data is transmitted to the cloud and reduces required bandwidth [12]. However, embedded systems have limited resources, such as memory or CPU power. Therefore, it is important that the ML algorithms are optimized for the embedded system. There are several tools and frameworks which can be used for this task [13, 11, 14]. In this paper we will report how much cloud storage, cloud computing cost and energy can be saved by running the ML algorithms directly on the embedded systems control unit, compared to running them on a cloud based solution.

2. Food Waste Management with Machine Learning

KITRO's IoT device is shown in figure 1. The key components are: a scale, a camera and an embedded control unit. The scale registers the increased weight when food gets thrown into the bin. The control unit activates the camera and stores the image locally on the device. Those images can then later be analysed using the embedded ML algorithms. Figure 2a and 2b show some example images.

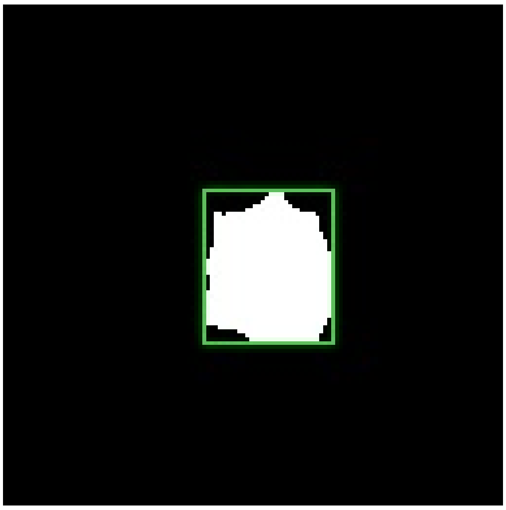
As the sample images indicate, food waste classification is a difficult computer vision task for several reasons: a) Low inter-class variance: Based on visual features only, it is difficult to differentiate between certain classes, i.e. different types of salad, b) High intra-class variance: Some food classes come in a wide variety of textures and colors if cooked or processed, c) Liquid food waste, e.g. soups and stews as well as soft food waste, e.g. chopped vegetables and salads, can hide visual features of other food classes [15]. ML algorithms, in particular deep learning approaches, seem most suitable for this challenging image processing task because they are able to generalize well [16] and can be fine-tuned to specific data of selected canteens or additional food classes [17]. However, the foundation of every deep learning algorithm is labeled data. KITRO has collected and labeled a large amount of food waste images that can be used as deep learning training data. The labels consist of food classes and the food weight. The labeled images are split into different food classes such as: *bread, banana, herb* etc. The images can contain one or more food or waste items. As an example the label of figure 2a is *herb* and the label of 2b is *banana* because those are the last added items. In addition, it can happen that the kitchen staff moves the bin or covers part or all of the bin. This makes food recognition additionally challenging. The ML algorithm has to detect and tag those error images such that actions can be taken to reduce the number of errors.

2.1. Machine Learning for Food Waste Analysis

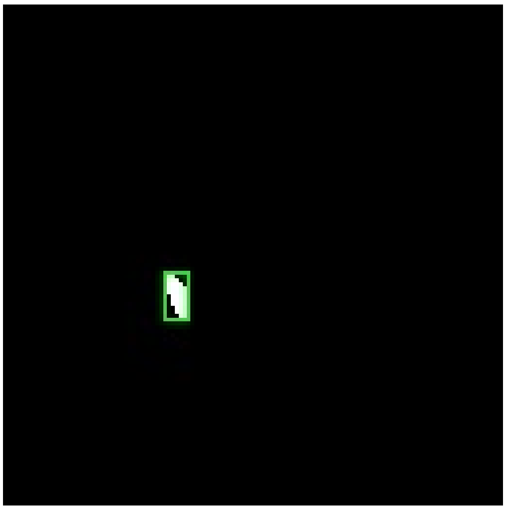
The ML algorithm has to solve two different tasks in order to analyse food waste. First it has to perform change detection, meaning that it has to detect the newly added food and mask that region in the input image. Then it has to classify the items within the masked region. Figure 2a resp. figure 2b show two example input images. Figure 2c and figure 2d show the corresponding change masks. The final pre-processed images are shown in figure 2e and figure 2f. The predicted labels for the food items are shown in the image captions.

(a) ML example input image with newly added food.

(b) ML example input image with newly added food.



(c) Detected newly added food item with change mask and bounding box.



(d) Detected newly added food item with change mask and bounding box.

(e) Final pre-processed image. ML label prediction: *herb*.

(f) Final pre-processed image. ML label prediction: *banana*.

Figure 2: Different machine learning processing steps.

3. TensorRT - Embedded Inference

We implemented the ML algorithms directly on the embedded device in order to obtain real-time predictions, increase privacy and reduce the required bandwidth and energy consumption. Running the ML algorithms in the cloud also offers advantages such as: scaling, online data analysis, data and processing in one place, simpler access to the data. We implemented the embedded ML models using the TensorRT (TRT) framework. To compare the performance to another widely used framework, they were also implemented with TensorFlow (TF). Both frameworks use the GPU of the embedded device to run ML inference. TensorRT includes a deep learning inference optimizer, a runtime code that delivers low latency and high throughput inference [14]. Those optimizations are required in embedded applications because embedded hardware has limited resources compared to cloud or desktop solutions. We have compared different metrics such as tera floating point operations per second (TFLOPS), gpu memory (GPUMEM), mean processing time (PT) (time required for the ML to process one image) and its standard derivation (std) across three frameworks and hardware platforms.

Table 1
Platform and framework performance comparison results. PT measurements were repeated for 100 times.

Platform	TFLOPs	GPUMEM	PT mean	PT std
Tesla P40 TF	11.76	24GB	0.15s	0.005
Embedded TF	0.472	4GB	1.65s	0.15
Embedded TRT	0.472	4GB	0.27s	0.0028

Using the TensorRT framework on the embedded device speeds up the inference time by a factor of 6 compared to standard TensorFlow. This speedup is crucial for real-time applications and decision making on the embedded system. Moreover, the embedded device has less FLOPs and memory available and is still able to run the food waste analysis ML in less than one second, which is comparable to the inference time on a Tesla P40. Furthermore, by only forwarding the labels to the cloud, a large amount of cloud resources can be saved. Table 2 shows the required memory for a typical image taken by the camera and the required memory for a .json text file which contains the results of the embedded ML algorithm.

Table 2
Example memory required to store results.

Data type	Required memory
.jpg image	1’881’180bytes
.json text	1’435bytes

If we assume that we process 1000 images per day, the following amount of cloud storage can be saved:

$$(1881180bytes - 1435bytes) * 1000 = 1.81GB \tag{1}$$

1.81GB of cloud storage is not that expensive today, with prices ranging from 0.005 to 0.021$/GB/Month [18]. However, those costs accumulate over time.

The last measurement was to estimate and compare the power consumption. The cloud implementation runs the ML algorithms on a Tesla K80 which has a power consumption of $300W$ [19]. Not every processing step in the cloud is done on the GPU and the Tesla K80 is not always running at 100%. Therefore, the average power was reduced by a factor of 2. If we again assume that we need to process 1000 images per day, the cloud implementation of the ML algorithms runs for $0.21h$ (including an estimation of time required to upload the data). The energy consumption per day on the cloud can then be estimated with:

$$300W * 0.5 * 0.21h = 31.5Wh \tag{2}$$

This energy consumption is only an estimation because we ignored the power consumption of uploading images and storing images etc. However, the estimation makes it possible to compare the cloud power consumption to the power consumption of the embedded device.

The embedded device had an average power consumption of $4.47W$ and $0.075h$ to process 1000 images. It's energy consumption (EC) can be estimated with:

$$4.47W * 0.075h = 0.34Wh \tag{3}$$

Table 3
Estimated energy consumption of running the ML algorithms in the cloud vs running them directly on the embedded system.

Platform	Estimated EC per day
Cloud	31.5Wh
Embedded	0.34Wh

Although the results of table 3 are estimated, they show the significant difference between the two solutions. By running the ML algorithms on an embedded device, energy consumption can be reduced by a factor of ~ 100. From a cost perspective the embedded device also offers advantages: Costs are significantly lower because the embedded device is already used in most of the IoT applications anyway and therefore only the power consumption adds additional costs. On the other hand, cloud instances have the advantage that they can be turned on and off on demand. The embedded system has to run all day long which again adds to power consumption and costs.

When designing an IoT solution, both methods - running ML in the cloud and running ML on the embedded system - should be considered, because both have different advantages and disadvantages. Depending on the specific use case, also a combination of the two approaches could be considered.

4. Discussion and Outlook

In this paper we presented how a smart IoT device can be used to analyse and classify food waste. ML algorithms enable the solution to detect and classify a broad range of different food items. By showing customers which kind and how much food they wasted, they can take measures to reduce the amount of wasted food and save costs. Furthermore, we showed the benefits of running ML algorithms directly on the embedded IoT device. Many IoT control devices today are capable of running small ML algorithms and are present in IoT solutions. Therefore, the only additional cost is the energy consumption which is significantly lower than running the same algorithms in the cloud. Another benefit is the real-time capability, the increased privacy, and the reduced memory requirement. Thus, depending on the use case, an embedded ML implementation is an interesting alternative to a cloud implementation.

Acknowledgments

This research was funded through an Innosuisse grant by the Swiss Innovation Agency (2019-2021).

References

[1] C. Beretta, S. Hellweg, Lebensmittelverluste in der schweiz: Umweltbelastung und vermeidungspotenzial, ETH Zürich, commissioned by the Swiss Federal Office for the Environment FOEN (2019). URL: https://www.newsd.admin.ch/newsd/message/attachments/58769.pdf.

[2] Auswirkungen von vermeidbaren lebensmittelverlusten auf die umwelt, https://www.bafu.admin.ch/bafu/de/home/themen/abfall/abfallwegweiser-a-z/biogene-abfaelle/abfallarten/lebensmittelabfaelle.html, 2021. Accessed: 2021-05-17.

[3] Nahrungsmittelverluste im detailhandel und in der gastronomie in der schweiz, Bericht in Erfüllung des Postulates 12.3907 von Isabelle Chevalley vom 28.9.2012, Accessed: 2021-05-17.

[4] K. Flanagan, A. Clowes, B. Lipinski, L. Goodwin, R. Swannell, Sdg target 12.3 on food loss and waste: 2018 progress report, An annual update on behalf of Champions 12 (2018). URL: https://champions123.org/sites/default/files/2020-09/champions-12-3-2019-progress-report.pdf.

[5] C. Reynolds, L. Goucher, T. Quested, S. Bromley, S. Gillick, V. K. Wells, D. Evans, L. Koh, A. Carlsson Kanyama, C. Katzeff, Åsa Svenfelt, P. Jackson, Review: Consumption-stage food waste reduction interventions – what works and how to design better interventions, Food Policy 83 (2019) 7–27. URL: https://www.sciencedirect.com/science/article/pii/S030691921830318X. doi:`https://doi.org/10.1016/j.foodpol.2019.01.009`.

[6] I. Hong, S. Park, B. Lee, J. Lee, D. Jeong, S. Park, Iot-based smart garbage system for efficient food waste management, The Scientific World Journal 2014 (2014). URL: https://www.hindawi.com/journals/tswj/2014/646953/.

[7] M. U. Sohag, A. K. Podder, Smart garbage management system for a sustainable urban life: An iot based application, Internet of Things 11 (2020) 100255. URL: https://www.sciencedirect.com/science/article/pii/S2542660520300901. doi:https://doi.org/10.1016/j.iot.2020.100255.
[8] S. Navghane, M. Killedar, V. Rohokale, Iot based smart garbage and waste collection bin, International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) 5 (2016) 1576–1578. URL: https://www.technoarete.org/common_abstract/pdf/IJERCSE/v5/i4/Ext_04798.pdf.
[9] Winnow vision, https://www.winnowsolutions.com/de/vision, Accessed: 2021-05-19.
[10] The leanpath food waste prevention platform, https://www.leanpath.com/, Accessed: 2021-05-19.
[11] R. David, J. Duke, A. Jain, V. J. Reddi, N. Jeffries, J. Li, N. Kreeger, I. Nappier, M. Natraj, S. Regev, R. Rhodes, T. Wang, P. Warden, Tensorflow lite micro: Embedded machine learning on tinyml systems, 2021. arXiv:2010.08678.
[12] R. Zingg, M. Rosenthal, Artificial intelligence on microcontrollers, in: Embedded World Conference 2020, Nürnberg, 25.-27. Februar 2020, 2020. URL: https://raw.githubusercontent.com/InES-HPMM/Artificial_Intelligence_on_Microcontrollers/master/Artificial_Intelligence_on_Microcontrollers.pdf.
[13] Ai expansion pack for stm32cubemx, https://www.st.com/en/embedded-software/x-cube-ai.html, Accessed: 2021-05-17.
[14] Nvidia tensorrt, https://developer.nvidia.com/tensorrt, Accessed: 2021-05-17.
[15] A. Mazloumian, M. Rosenthal, H. Gelke, Deep learning for classifying food waste, arXiv preprint arXiv:2002.03786 (2020). URL: https://arxiv.org/abs/2002.03786.
[16] K. Kawaguchi, L. P. Kaelbling, Y. Bengio, Generalization in deep learning, 2020. arXiv:1710.05468.
[17] C. Käding, E. Rodner, A. Freytag, J. Denzler, Fine-tuning deep neural networks in continuous learning scenarios, in: Asian Conference on Computer Vision, Springer, 2016, pp. 588–605. URL: https://link.springer.com/chapter/10.1007/978-3-319-54526-4_43.
[18] Cloud storage pricing comparison, https://www.backblaze.com/b2/cloud-storage-pricing.html, 2021. Accessed: 2021-05-19.
[19] NVIDIA, Tesla k80 gpu accelerator board specification, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-product-literature/Tesla-K80-BoardSpec-07317-001-v05.pdf, 2021. Accessed: 2021-05-19.

A. Online Resources

An example workflow for custom TensorRT models is available on

- GitHub