PhasmaFOOD - A miniaturized multi-sensor solution for rapid, non-destructive food quality assessment

Benedikt Groß[1], Susanne Hintschich[2], Milenko Tošić[3], Paraskevas Bourgos[4], Konstantinos Tsoumanis[4], and Francesca Romana Bertani[5]

[1] Freie Universität Berlin, Institut für Informatik
Heisenberg Communications and Information Theory Group
Takustraße 9, 14195 Berlin (Germany)
[2] Fraunhofer Institut für Photonische Mikrosysteme IPMS
Maria-Reiche-Str. 2, 01109 Dresden (Germany)
[3] Vizlore Labs Foundation
Braće Ribnikar 56, 21000 Novi Sad (Serbia)
[4] Wings ICT Solutions
189 Siggrou Avenue, 17121 Athens (Greece)
[5] CNR-IFN Institute for Photonics and Nanotechnologies
Via Cineto Romano 42, 00156 Rome (Italy)

Abstract PhasmaFOOD is a H2020 project with the goal of building a miniaturized, smart multi-sensor food scanner. Equipped with a NIR sensor, a UV-VIS sensor and a RGB camera it aims to be a portable, highly versatile solution for various food safety issues, ranging from aflatoxin detection in grains and nuts, over shelf-life prediction in meats and fish to detection of adulteration in meat, edible oils and alcoholic beverages. The unique combination of sensors, operation via a smartphone application and sophisticated data analysis methods offer the possibility of rapid, non-destructive measurements that can - in contrast to costly and slow laboratory instruments - be applied at every stage of the production chain, from farm to fork. After a brief introduction of the PhasmaFOOD system architecture the data analysis approach, especially the image analysis, based on dictionary learning is explained in detail.

Keywords: PhasmaFOOD, food scanner, optical sensing, spectroscopy, learning, image classification, image compression.

DOI: 10.58895/ksp/1000087509-10 erschienen in:

OCM 2019 - 4th International Conference on Optical Characterization of Materials, March 13th – 14th, 2019, Karlsruhe, Germany

DOI: 10.58895/ksp/1000087509 | https://www.ksp.kit.edu/site/books/m/10.58895/ksp/1000087509/

1 Introduction

PhasmaFOOD (www.phasmafood.eu) is a H2020 funded project of the European Union with a strong consortium of 9 stakeholders with expertise in food safety, spectroscopy, hardware and software development and machine learning, namely Intrasoft International S.A., Wings ICT Solutions, Ltd, VizLore Labs Foundation, RIKILT - Wageningen Reasearch, Agricultural University of Athens, Italian National Research Council, University of Rome Tor Vergata, Fraunhofer IPMS and Freie Universität Berlin. The objective of the PhasmaFOOD project is to develop a miniaturized, portable, smart multi-sensor food scanner. The system comprises a miniaturized, portable device integrating three different sensors and various light sources and a distributed software architecture. The PhasmaFOOD software architecture consists of a programmable micro-controller that steers the sensors in the portable device, a mobile app that serves as an interface to the user and is used to conduct measurements, calibrate the device and present recommendations based on the decision making in the cloud platform. The unique combination of three different optical sensors, a Near Infra-Red (NIR) spectrometer, a UltraViolet (UV) - Visible (VIS) spectrometer, operable in reflectance and fluorescence mode, a high resolution color imaging system, and sophisticated machine learning algorithms offer the potential to cover a wide range of applications in food safety.

The PhasmaFOOD solution is designed to meet three major requirements: **Portability**: the spectrometer is hand-held and can be configured on the mobile app. Hence, it works in various environments where food is sold or processed. **Versatility**: The specific combination of sensors (NIR spectrometer, UV-VIS spectrometer and CMOS camera) covers a spectal range from 400nm to 1900 nm, and, via the camera provides also textural information about the sample at hand. Hence, a wide range of food types and use cases can be targeted. **Fast, non-destructive predictions**: The three optical sensors work without the need to damage the product under investigation and deliver a measurement instantly. Therefore, it is well suited to time-critical problems like the shelf-life prediction of raw meat or fish, where waiting several days for results from laboratory measurements is not an option. The targeted use cases include:

1. Detection of mycotoxins in various grains and nuts. Aflatoxin detection.
2. Detection of (early signs of) spoilage in fruits, vegetables, meat and fish. Shelf-life prediction.
3. Detection of food fraud in alcoholic beverages, oil, milk powder and meat.

2 PhasmaFOOD system architecture

The PhasmaFOOD system consists of a miniaturized sensing device, a mobile application and the PhasmaFOOD cloud platform. The sensing device is used to take measurements of food samples with its integrated optical sensors. The sensing system is connected via bluetooth to a smart mobile device, such as a tablet or mobile phone, from which it is operated via the PhasmaFOOD mobile application. The mobile app receives data from the sensing device and forwards them to the cloud platform. On the PhasmaFOOD cloud platform the data is stored in the database and decision making algorithms are applied to incoming measurements. The predictions obtained from the use-case specific analysis algorithms are sent back to the mobile application and presented to the user.

2.1 Sensing device

The PhasmaFOOD sensing device is a portable multi-sensor device, comprising the sensing sub-unit, in which a NIR spectrometer, a UV-VIS spectrometer and a CMOS camera, illumination units and driving boards for all components are located, and the electronics sub-unit, a custom-built microcontroller, equipped with a recharchable battery, several communication interfaces (USB, BLE, WiFI), external memory (microSD) and additional sensors (inertial measurement unit, temperature sensor). The electronics sub-unit is able to configure the sensing sub-unit, read-in the raw sensor data, perform preprocessing operations and communicate with the mobile app. Due to the integration of an ARM processor, RAM and a FPGA unit even advanced processing of sensory data can be performed on the sensing device itself. The details of the electronics sub-unit can be found in [1] The sensing sub-unit (shown in figure 10.1(a))and the electronics sub-unit are mounted in a 3D-printed housing as shown in figure 10.1(b).

(a) PhasmaFOOD sensing sub-unit

(b) PhasmaFOOD portable multi-sensor device

Special emphasis was given to a modular design in order to enable the replacement of sensing or lighting components. For the current prototype the following sensors were chosen: A miniaturized NIR spectrometer by Fraunhofer IPMS [2] (spectral range: $1000 - 1900nm$, size: $17 \times 12 \times 16mm^3$), the Hamamatsu C12880MA UV-VIS spectrometer (spectral range: $340 - 850nm$, size: $20.1 \times 12.5 \times 10.1mm^3$), and the Ximea MU9PC-MH CMOS camera (resolution: $5MP$, size: $15 \times 15 \times 8mm^3$). The range of the two spectrometers covers the entire visible and the near infra-red spectrum and is supported by a RGB camera to include spatial information as well. The sensors are accompanied by various lighting units, i.e. white LEDs, a NIR and a UV lightsource. Hence, the device is able to record a NIR spectrum, a fluorescence measurement, a visible reflectance spectrum of the sample under UV illumination and a RGB camera image.

2.2 Mobile application

The device is operated by the user via the PhasmaFOOD mobile app. In the mobile app the user selects one of the pre-defined use cases and the food type under consideration. The app then guides the user through the measurement process, displays the data and presents analysis results in comprehensible form. Expert users can also configure the sensing device for new use cases, adjust the lighting and tweak the parameters of the sensors, such as integration time, number of internal measurements etc. through the app. The measurements received from the device via BLE, together with additional input from the user are bundled in one json object and sent to the cloud platform for further processing and analysis. The current status of the mobile application is described in [3].

2.3 Cloud platform

The cloud platform is the focal point of the PhasmaFOOD system. Running on two virtual machines, it hosts the data warehouse, the rule engine for decision making, the web dashboard and the machine learning 'playground', a tool for configuring new machine learning pipelines for each available dataset. For each use case exists a labelled dataset in the data warehouse, on which machine learning algorithms for each sensor are trained. These datasets are produced by experts in food chemistry laboratories and contain measurements with additional instruments that give ground truth values for the quantities of interest that the PhasmaFOOD system attempts to predict, i.e. aflatoxin contamination for use case 1, microbia counts and age of the samples for use case 2 and information on adulterands for use case 3. Based on these decision making algorithms, the rule engine outputs a verdict on the food quality of an incoming measurement, which is presented to the user in the mobile app. Details on the cloud functionality can be found in [3].

3 Data analysis strategy

The different steps that transform the raw measurements into a food quality verdict are distributed across the PhasmaFOOD system. Simple preprocessing of the sensor data is done in the electronics subsystem of the PhasmaFOOD device, whereas the extraction of high level features and the final prediction is currently done on the cloud platform. A redistribution of these decision making steps to the mobile app or even the device to optimize the use of resources will be subject of further investigation in the last stage of the PhasmaFOOD project. The decision making is based on supervised learning methods. While simple preprocessing steps are the same for all food types under consideration, the computation of expressive features and the best prediction algorithm for a certain quantity of interest vary with use case and food type. Hence, for each labelled dataset corresponding to a use case and food type a variety of models are trained and evaluated to find a suitable analysis strategy. The data recorded with the spectrometers only amount to a few kilobytes. Thus, there is no need to perform compression on the embedded device. The image data on the other hand requires special treatment to reduce the traffic over the BLE connection from the device to the mobile app and subsequently to the cloud.

3.1 Structured dictionary learning

It is desirable to reduce the size of the images (ca. 15Mb) significantly thereby retaining the significant information for later classification. Since natural images are highly redundant, a common idea for lossy compression is to represent the image as a linear combination of suitably chosen dictionary atoms. General purpose lossy compression algorithms like JPEG or JPEG2000 use fixed dictionaries such as cosine atoms or wavelet atoms, respectively that are well suited to represent natural images. While these algorithms focus entirely on good reconstruction of all natural images, measured by human perception, the goal for image compression in the PhasmaFOOD project is different. Each use case and food type produces a very narrow class of images that look very similar, with subtle variations due to spoilage of aged or adulterated food samples. These variations must be captured for further analysis at the same time reducing the ammount of dat to be transfered. Finally, since the compression/feature extraction algorithm runs on the embedded device, it should be fast and with low computational complexity. Therefore, the idea is to learn a data-adapted dictionary that focusses on sparse and discriminative encoding. The time-consuming learning part can be done offline on the cloud platform. For the encoding with the learned dictionary fast algorithms exist that can be run on the device as shown in figure 3.1. Dictionary

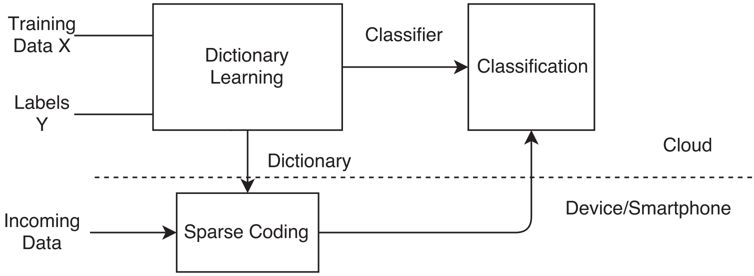

Figure 10.1: Compression scheme.

learning [4] has proven a successful technique for a variety of imaging task, such as sparse coding, denoising or image inpainting. In its basic form, it solves the problem

$$\min_{D,A} \|X - DA\|_2^2 + \lambda\|A\|_1 \quad \text{subject to } \|d_l\|_2 = 1 \quad \text{for } l = 0, 1, 2, \ldots, \tag{10.1}$$

where $D \in \mathbb{R}^{n \times N}$ is the dictionary to be learned, $X \in \mathbb{R}^{n \times m}$ is the matrix of training samples and $A \in \mathbb{R}^{N \times m}$ the matrix containing the codes. The regularization term $\lambda\|A\|_1$ enforces sparsity in the codes and the columns d_l of the dictionary are constrained to have unit norm to avoid scaling ambiguities. The problem can be solved by iteratively alternating between computing the codes A thereby keeping D fixed, i.e. sparse coding, and updating the dictionary atoms to better represent the data for fixed sparse codes. Additional terms can be incorporated in the objective function to induce task-specific properties of the solution. In [5] a classification loss was added to render the problem supervised, in [6] label information was included via a Fisher discrimination criterion. In [7] the dictionary was structured into a common and class-specific parts, which was used in [8] for fine-grained image classification. This approach is adopted here and combined with incoherence promoting terms [9] that stabilize the sparse coding and encourage the sub-dictionaries to encode different information. Let the dictionary be partitioned as $D = [D_0, D_1, \ldots, D_L]$, where $D_0 \in \mathbb{R}^{n \times N_0}$ is the common dictionary and $D_j \in \mathbb{R}^{n \times N_j}$ for $j = 1, \ldots, L$ are the class specific dictionaries. As another minor modification to the basic problem 10.1, instead of the l_1 regularization of A, here the number of nonzeros is constrained directly to be $\leq s$. Each sample x_i with corresponding label y_i is encoded only using D_0 and D_{y_i}. With the aforementioned incoherence terms the objective becomes

$$f(D,A) = \|X - DA\|_2^2 + \sum_{j=0}^{L} \mu_j \|D_j^T D_{-j}\|_2^2 + \sum_{j=0}^{L} \eta_j \|D_j^T D_j - I_{N_j}\|_2^2, \quad (10.2)$$

where D_{-j} denotes the dictionary composed of all but the jth sub-dictionary and I_N is the $N \times N$ identity matrix. Let $\mathcal{S}_j$ denote the indices of dictionary atoms belonging to the jth sub-dictionary. Then the discriminative encoding property of the codes can be expressed as the constraint

$$a_i\Big(\bigcup_{k \notin \{0,j\}} \mathcal{S}_k\Big) = 0 \text{ if } y_i = j. \quad (10.3)$$

The sparse coding step for a training sample x_i with label y_i takes the form

$$\min_a \|x_i - [D_0, D_{y_i}]a\|_2^2 \quad \text{subject to } |a|_0 \leq s, \quad (10.4)$$

which can be solved efficiently by orthogonal matching pursuit (OMP) [10]. OMP only needs to compute inner products between the sample and the dictionary atoms and is hence suited to run on the embedded device, which has only limited capacities.

For the dictionary update step the algorithm in [9] is employed. Denote by X_j the samples in class j and A_j the corresponding sparse codes. Set

$$Z_j = X_j - D_0 A_j(\mathcal{S}_0) \qquad \text{for } j = 1, \ldots, L,$$
$$Z_0 = X - [D_1, \ldots, D_L]A(\mathcal{S}_{-0}).$$

Computing the derivative of the objective with respect to a sub-dictionary D_j and setting the result to 0, $\frac{\partial f(D_j)}{\partial D_j} = 0$ leads to a Sylvester-type matrix equation

$$PD_j + D_j Q = R \quad (10.5)$$

with the matrices

$$P = \mu_j D_{-j} D{-j}^T,$$
$$Q = A_j(\mathcal{S}_j)A_j(\mathcal{S}_j)^T + \eta_j D_j^T D_j - I_{N_j},$$
$$R = Z_j A_j(\mathcal{S}_j)^T.$$

The Sylvester equation could be solved by the Bartels-Stewart algorithm [11], if Q and R were independent of D_j. Following [9], an approximate solution to equation 10.5 can be obtained by intializing $D_j^0 = D_j$ and then solving $PD_j + D_jQ^t = R^t$ for a few iterations, where Q^t and R^t are computed using D_j^{t-1} (normalized to have unit norm columns) and the updated sparse codes.

To apply the described algorithm the raw image is first scaled to the range $[0, 1]$ in each channel, then a square region of interest is extracted to exclude the edges of the sample holder. The selcted region is divided into non-overlapping patches of shape $(p \times p \times 3)$, which then serve as input to the algorithm, resulting in dictionary atom size $3p^2$. The sparse codes of an image's patches can be quantized and entropy coded to further reduce the size, if needed. In the cloud, an SVM is trained on the sparse codes of the training samples, resulting in a verdict for each image patch.

4 Discussion and conclusion

Although the literature on dictionary learning suggests good results in terms of compression rates and high accuracies have been achieved in classification tasks using structured dictionaries, the performance of the presented approach for the specific application in the context of PhasmaFOOD remains to be thoroughly investigated once enough labelled data has been collected. Furthermore, image data may not in all use cases contain information about the state of the food samples under consideration. This must be taken into account in data fusion strategies. In case one sensor does not contain useful information, a high level fusion strategy, i.e. combining the single predictions together to get a final verdict can simply assign a low weight to the useless sensor. In low level data fusion approaches this useless sensor might severly distort the classifier, unless a very strict feature selection method is used.

5 Summary

In this article the PhasmaFOOD system architecture was described. An approach to image compression and feature extraction based on learning a structured dictionary from labelled training data was presented in detail. The training of this dictionary is time consuming, but can be done offline on powerful computers, whereas the encoding of an image with it can be done on an embedded device efficiently by using OMP.

References

1. PhasmaFOOD, "Deliverable report d5.2: Hardware processing platform, storage and communication units integration -v1," 2018.
2. T. Pügner, J. Knobbe, H. Grüger, and H. Schenk, "Realization of a hybrid-integrated mems scanning grating spectrometer," in *Next-generation spectroscopic technologies V*, vol. 8374. International Society for Optics and Photonics, 2012, p. 83740W.
3. PhasmaFOOD, "Deliverable report d4.1: First implementation of software for phasmafood smart food analysis system," 2018.
4. M. Aharon, M. Elad, A. Bruckstein *et al.*, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, p. 4311, 2006.
5. J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, "Supervised dictionary learning," in *Advances in neural information processing systems*, 2009, pp. 1033–1040.
6. M. Yang, L. Zhang, X. Feng, and D. Zhang, "Sparse representation based fisher discrimination dictionary learning for image classification," *International Journal of Computer Vision*, vol. 109, no. 3, pp. 209–232, 2014.
7. S. Kong and D. Wang, "A dictionary learning approach for classification: separating the particularity and the commonality," in *European Conference on Computer Vision*. Springer, 2012, pp. 186–199.
8. S. Gao, I. W.-H. Tsang, and Y. Ma, "Learning category-specific dictionary and shared dictionary for fine-grained image categorization," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 623–634, 2014.

9. M. Nejati, S. Samavi, N. Karimi, S. M. R. Soroushmehr, and K. Najarian, "Boosted dictionary learning for image compression." *IEEE Trans. Image Processing*, vol. 25, no. 10, pp. 4900–4915, 2016.
10. J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on information theory*, vol. 53, no. 12, pp. 4655–4666, 2007.
11. R. H. Bartels and G. W. Stewart, "Solution of the matrix equation ax + xb = c [f4]," *Commun. ACM*, vol. 15, no. 9, pp. 820–826, Sep. 1972.