Gerina *et al. Hum. Cent. Comput. Inf. Sci.* (2020) 10:27
https://doi.org/10.1186/s13673-020-00235-9

Human-centric Computing and Information Sciences

RESEARCH Open Access

Recognition of cooking activities through air quality sensor data for supporting food journaling

Federica Gerina, Silvia M. Massa, Francesca Moi, Diego Reforgiato Recupero and Daniele Riboni*

*Correspondence:
riboni@unica.it
Dept. of Mathematics and Computer Science, University of Cagliari, via Ospedale 72, 09124 Cagliari, Italy

Abstract

Unhealthy behaviors regarding nutrition are a global risk for health. Therefore, the healthiness of an individual's nutrition should be monitored in the medium and long term. A powerful tool for monitoring nutrition is a food diary; i.e., a daily list of food taken by the individual, together with portion information. Unfortunately, frail people such as the elderly have a hard time filling food diaries on a continuous basis due to forgetfulness or physical issues. Existing solutions based on mobile apps also require user's effort and are rarely used in the long term, especially by elderly people. For these reasons, in this paper we propose a novel architecture to automatically recognize the preparation of food at home in a privacy-preserving and unobtrusive way, by means of air quality data acquired from a commercial sensor. In particular, we devised statistical features to represent the trend of several air parameters, and a deep neural network for recognizing cooking activities based on those data. We collected a large corpus of annotated sensor data gathered over a period of 8 months from different individuals in different homes, and performed extensive experiments. Moreover, we developed an initial prototype of an interactive system for acquiring food information from the user when a cooking activity is detected by the neural network. To the best of our knowledge, this is the first work that adopts air quality sensor data for cooking activity recognition.

Keywords: IoT, AI, Machine learning, Activity recognition, Healthcare

Introduction

The 2018 Global Nutrition Report.[1] of the World Health Organization (WHO) reveals that malnutrition affects, in different forms, every country of the world. Malnutrition determines more health issues than any other cause, and progress towards better nutrition is still too slow. In particular, in developed countries, overweight and obesity in adults are a cause of several non-communicable diseases, including diabetes, heart disease, stroke, different types of cancer, musculoskeletal disorders, and respiratory symptoms.[2] Micronutrient deficiencies are also cause of severe health issues, such as anaemia.

[1] https://www.who.int/nutrition/globalnutritionreport/en/.

[2] https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight.

In the 2018 Report, the WHO envisions five critical steps to speed up progress toward better nutrition. Among them, the WHO states the need to prioritize gathering and analysis of diet data. Diet data analysis is of foremost importance to evaluate the healthiness on an individual's nutrition, and for setting up interventions when necessary. While several systems have been proposed in the literature to monitor food intake, which are reviewed in "Related work" section, at the time of this writing, a complete solution to acquire diet data for long-term data analysis is still missing. In particular, as we explain in the following, many existing solutions interfere with the normal activities of the user, and are unsuitable for elderly or frail people, while other ones (including those based on cameras) determine serious privacy issues.

A powerful tool for acquiring diet data is *food journaling*, which consists of filling a diary of eaten food and its quantity at each meal [11]. Of course, manually keeping a food diary in the long term is tedious and impractical, since in many cases the diary annotation interferes with the current activity. In order to assist the individual in filling a food diary, different solutions have been proposed in the last years, which exploit mobile apps and smartphone sensors such as the camera or microphone [3, 9, 22, 38]. Mobile apps for food journaling may reduce the burden of data entry [5]; however, as discussed in "Related work" section, they still require considerable user's effort, and are often unsuitable for frail people, including the elderly.

In this paper, we propose to exploit artificial intelligence and innovative devices, such as indoor air quality monitors, to help addressing this challenge. Indeed, we believe that these technologies, possibly coupled with automatic dialog systems, may provide adequate solutions for unobtrusive and privacy-conscious context recognition, as well as user engagement for supporting diet data acquisition on the long term. In particular, in this paper we concentrate on the technical solutions and evaluation of the modules for recognizing cooking activities. Advanced solutions for conversational interfaces [7], human–computer interaction [27], usability [34], and diet analysis [17], including calorie count [29] and adaptive interfaces for supporting behavior change [25], will be addressed in future work.

Our system relies on data captured by a commercial air quality sensor to recognize food preparation using a deep learning approach. The air quality sensor that we adopt in this work can monitor several parameters, including temperature, humidity, carbon dioxide, volatile organic compounds, particulate matter, nitrogen dioxide, carbon monoxide, and ozone. It takes readings at each minute and sends them to the cloud. The system exploits open APIs to continuously query the measured data. Based on the continuous stream of sensor data, the system extracts features considering a temporal window of 30 min. Features are carefully engineered to capture the trends of food preparation, which include typical patters regarding changes in both gas levels and environmental parameters, due to the activity in the kitchen and to the use of tools such as the oven or gas stove. Those features are used by a deep neural network to recognize the preparation of food in real time. Our network is composed of four fully connected layers, and adopts specific solutions to prevent over-fitting. The network has been trained using a large dataset of cooking activities acquired by several participants in real-world homes under different conditions for more than 8 months in total. Compared to other cooking recognition systems based on cameras or other kinds of sensors, our data acquisition

system is unobtrusive and privacy-conscious. In fact, we do not rely on cameras, because several people may perceive them as very intrusive in terms of privacy. Moreover, we rely on a single plug-and-play sensor that does not require any kind of manual calibration or setup. In order to demonstrate the operation of our system, we developed an initial prototype in which the cooking recognition system activates a social robot to interactively acquire food information from the user; however, our system could be coupled with other interactive systems, such as digital personal assistants or dialog systems.

Currently, our system is addressed to frail people that live alone and take most meals at home, since this is a typical situation for many elderly people. However, to support other categories of users, our system may be extended with mobile solutions to fill the food diary when the user takes food outside the home. Being based on air quality monitors, our current system is well suited to recognize the preparation of warm meals. In order to recognize the preparation of cold meals, our system can be seamlessly extended with other sensors attached to kitchen appliances such as the fridge or to kitchen furniture. Those additional sensors would also be useful for improving the recognition rates of warm meal preparation. The support of multi-inhabitant scenarios will also be the subject of future investigation. Our work lies within the PhilHumans project,[3] a H2020 Innovative Training Network which is centered on the employment of artificial intelligence technologies to establish user interactions with their personal health devices in the most effective way.

More in detail, the contributions of this paper are the following:

- We present a novel technique for detecting food preparation activities in the home exploiting air quality sensor data and deep learning. In terms of unobtrusiveness and privacy-consciousness, our technique has clear advantages with respect to solutions based on wearable sensors or cameras.
- We introduce a novel system for food journaling addressed to frail people living alone. With respect to existing solutions, our system may engage the user based on automatic recognition of cooking activities and interactive food journaling acquisition.
- We release in a public repository the annotated sensor data that we collected, as well as the deep neural network we deployed for the activity recognition task.
- We have implemented a first prototype of part of the system using commercial sensors and a social robot.
- Within the food journaling area, we discuss the challenges we had to face and that we addressed using innovative technologies, and many others that we still need to face.

The rest of this paper is organized as follows. "Related work" section discusses background work related to the monitoring of nutrition behavior, including sensor-based recognition of cooking activities, and natural language processing methods for food journaling data acquisition. How the air quality sensor data have been acquired and how features have been engineered on top of them to be fed to a deep neural network is covered within "Acquisition and processing of air quality sensor data" section. The

[3] http://philhumans.eu/.

experimental campaign we have carried out, along with the results we have obtained, are presented in "Experimental evaluation" section. "Use case on a robotic platform" section illustrates the prototype we have developed for the proposed use case. Finally, "Conclusion and future work" section concludes the paper with final remarks and reports the future directions where we are headed.

Related work

According to the World Health Organization, unhealthy behaviors regarding nutrition and physical activity are a global risk for health. In particular, a healthy diet is of paramount importance for avoiding all forms of malnutrition, and it is a key factor for protecting against noncommunicable diseases such as diabetes, heart disease, stroke and cancer.[4] While those health issues are prominent in adult and elderly people, healthy nutritional practices should be followed during the whole course of life, starting from infancy. Unfortunately, despite all the effort put in the last decades by health education for improving dietary habits, good dietary practices are still neglected by relevant parts of the population. Hence, there is a growing need for novel technologies that can assist the individual in following good nutrition practices [4].

Of course, the healthiness of an individual's nutrition must be monitored in the medium and long term. Hence, a powerful tool for monitoring the nutrition behavior is a *food diary*; i.e., a daily list of food taken by the individual, together with portion information. Accurate food journaling can support both self-management of nutrition routines [20], and assessment by practitioners [11]. In particular, for diabetes patients, it is important to analyse the daily food intake and compare it with symptom progression [21]. A four-center randomized trial about weight loss maintenance shows that patients compiling a food diary lost twice as much weight than patients that kept no journaling [15]. Moreover, in the short term, food journaling may increase real-time awareness and mindfulness, avoiding the consumption of unhealthy food [33].

Traditional methods for keeping food diaries rely on interviews and questionnaires to assess the eating routines of patients [23]. More recently, several food diary apps have been proposed to support the individual in self managing his/her food diary be means of mobile devices [5]. Usually, those apps let the user manually fill in the kind of food and the portion information; the app queries a database and returns the calories count. Food information and calories counts are stored locally or on the cloud for long term monitoring [9]. Different solutions have been proposed to alleviate the burden of manually entering food and portion information in those apps. A popular direction consists in the use of food pictures and computer vision tools for semi-automatic food journaling. In particular, Zhu et al. propose the use of computer vision tools and pictures taken before and after food consumption to accurately recognize the kind of food and estimate the eaten quantity [38]. Sen et al. [31] propose a smartwatch-based system to detect eating gestures, and recognize food through pictures and computer vision software. A text-based conversational agent is proposed in [6] to improve nutritional lifestyle. Other solutions rely on automatic classification of chewing sound [3], recognition of eating

[4] https://www.who.int/news-room/fact-sheets/detail/healthy-diet.

moments trough analysis of heart rate and activity patterns [26], or scanning of grocery receipts [22]. Other systems, including the one proposed by Chi et al. [8], provide accurate calories counts using a combination of cameras, connected kitchen scales, and food databases. While these and similar systems reduce the burden of manual entry, they still require considerable user's effort.

Unfortunately, keeping a constant journal of meals is considered tedious and time-consuming by many individuals, and food diaries are rarely taken accurately in the long term [10]. Moreover, duration of food journaling practice is only a marginal determinant of nutrition healthiness [2], while other factors including the user's engagement play a key role. For this reason, conversational interfaces, possibly coupled with social robots when appropriate, may provide an effective solution to engage the user in keeping a food journal on the long term. An advanced conversation between the conversational agent and the user would require the employment of complex Natural Language Processing (NLP) techniques. In the literature, several novel works have been proposed that employ cutting-edge approaches to address that challenge. For example, authors in [24] centered on the use case of people teaching a robot about objects and tasks in its environment via unconstrained natural language. They designed statistical machine learning approaches to allow robots to gain knowledge about the world from interactions with users, while simultaneously acquiring semantic representations of language about objects and tasks. Moreover, authors in [14] present an adaptive and interactive dialogue system to exchange a chat with a user using personal information stored in his/her user profile. NLP is used to extract user's basic information, hobbies and interests for building a rich user profile. The user profile is continuously updated whenever new information is extracted in subsequent dialogues. Furthermore, NLP technologies have been recently employed for the design of virtual assistants, whose adoption has enabled several changes in today's life. Google Assistant,[5] Apple's Siri,[6] Microsoft's Cortana,[7] Amazon's Alexa,[8] Wit.ai[9] and Snips.ai,[10] an open source and privacy oriented solution, are all well-known examples of digital voice assistants available on the market today. Such technologies can be used to access apps, services, and software, reading out the daily news, setting timers, or tapping into music playlists, and control IoT products and sensors.

Another line of research consists in monitoring the activities of cooking and eating using machine learning methods and data acquired from cameras or sensors. Rohrbach et al. applied two different approaches to the recognition of cooking activities from data acquired by fixed cameras [28]. Their experimental results showed that the approach using holistic video features outperforms one based on articulated pose tracks, even though the latter is more effective in the recognition of fine-grained actions. Other works rely on head-mounted cameras for egocentric vision. In particular, Kazakos et al. propose the use of temporal binding methods to fuse audio and video signals from

[5] https://assistant.google.com/.

[6] https://www.apple.com/siri/.

[7] https://www.microsoft.com/en-in/windows/cortana.

[8] https://developer.amazon.com/alexa.

[9] https://wit.ai/.

[10] https://snips.ai/.

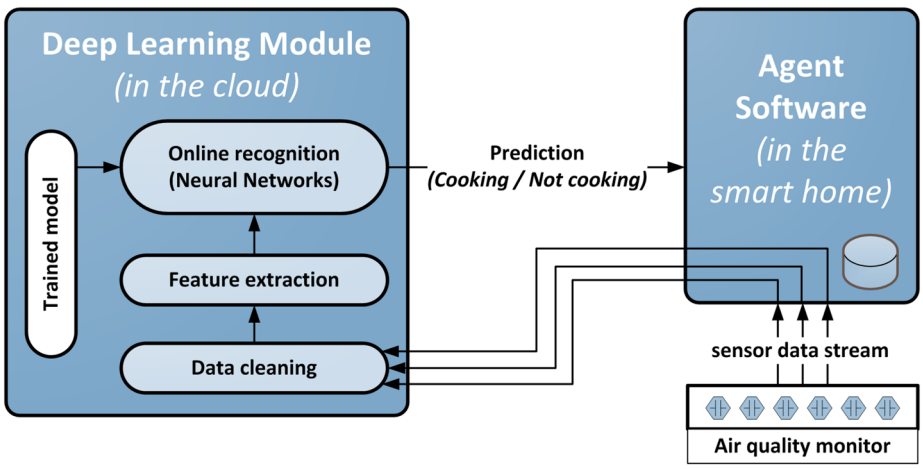

Fig. 1 Air quality sensor data acquisition and processing

egocentric cameras for recognition of different kitchen activities, including cooking [16]. However, the use of video and audio recording poses obvious issues in terms of privacy. In general, systems based on sensors are considered less invasive in terms of perceived privacy. Hence, other techniques rely on body-worn or environmental sensors for activity recognition. In the context of the SPHERE project [39], Yordanova et al. propose the use of computational state space models (i.e., probabilistic models combining symbolic representation with probabilistic reasoning) for recognizing cooking activities based on data acquired from different sensors in a smart kitchen [36, 37]. In particular, the authors rely on temperature, humidity, light/noise/dust levels, individual's motion, and usage of certain objects, water, and electricity. With their method, the authors achieve good recognition rates. However, installation and maintenance of many heterogeneous sensors may incur relevant costs. In this paper, we take the same approach, but we pursue the use of a single sensing device, which can be installed from scratch in a home with essentially no effort. Of course, the accuracy of our system could be increased by considering additional sensors.

Acquisition and processing of air quality sensor data

In this section, we explain how we acquire and process air quality sensor data in order to recognize the preparation of food. The diagram in Fig. 1 shows our framework for data acquisition and processing. An indoor air quality monitor deployed in the kitchen is in charge of providing a stream of real-time sensor data to a DATA CLEANING module. That module performs data preprocessing to eliminate possible errors in sensor readings. Then, the data is passed to a FEATURE EXTRACTION module, that builds feature vectors based on statistics computed on the current and past data. The feature vector is passed to the ONLINE RECOGNITION module, which uses an Artificial Neural Networks classifier to detect whether the user at home is cooking or not. The Neural Network is trained in advance using a labeled training set of sensor data acquired during cooking and non-cooking activities. Finally, the prediction (either *cooking* or *not cooking*) is communicated

to a robot, who is in charge of interacting with the user in order to interactively collect his/her food diary.

Sensor data acquisition

As shown in our experiments, reported in "Experimental evaluation" section, the act of preparing food determines relevant changes in the air quality of the cooking area. In particular, the use of a gas cooker determines an immediate increase of the carbon dioxide (CO_2) level in the kitchen. The preparation of certain kinds of food generates fumes containing different levels of volatile organic compounds (VOCs) [32]; the increase of VOC levels is particularly evident when certain foods are prepared, such as meat and fish. Similarly, cooking certain kinds of food determines the emission of particulate matter (PM); i.e., microscopic matter suspended in the air. The concentration and size of particulate matter is determined both by the cooking style (roasting, frying...) and by the used ingredients [1]. Natural gas stoves also emit other gases, such as NO_2, in the kitchen. Moreover, when cooking takes place, the environmental parameters of the kitchen are affected both in terms of temperature and humidity values.

Nowadays, indoor air quality monitors are becoming popular, due to their low cost and increased attention of people to the healthiness of indoor air. Our intuition is that it is possible to exploit off-the-shelf indoor air quality sensors in order to recognize food preparation activities by applying machine learning techniques to the sensor data stream. The advantage of this solution with respect to other ones based on cameras or environmental sensors is that the indoor air quality sensor is unobtrusive and requires negligible installation effort. Moreover, it is obviously more privacy-conscious than solutions based on microphones and cameras.

At the time of writing, different indoor air quality monitors are available on the market. These devices mainly differ from one another with respect to the kind of monitored parameters, detection frequency, form factor, network interfaces, presence of open APIs, and cost. For recognizing food preparation activities, we target a device having the following characteristics:

- it is able to monitor at least the following parameters: temperature, humidity, carbon dioxide, volatile organic compounds, particulate matter;
- it provides a detection frequency of at least one sensor reading per minute;
- for ease of installation, it provides a wireless network interface and electrical connection to avoid battery exhaustion;
- it provides open APIs for acquiring the sensor data in real time.

Among several indoor air quality meters currently available on the market, we chose the uHoo© device,[11] which provides all the desired characteristics mentioned above. Figure 2 shows a uHoo device used in our experimental setup. In addition to the mentioned parameters, the uHoo device also measures nitrogen dioxide, carbon monoxide, ozone, and air pressure. It provides sensor readings at 1 min frequency, and the data can be downloaded either in batch from a smartphone app, or in real time thanks to open APIs.

[11] https://uhooair.com/.

Fig. 2 An air quality monitor used in our experimental setup

Data cleaning

In general, sensor data are affected by a relevant level of noise. Hence, before being used, the raw sensor readings must be preprocessed to reduce the noise, which could negatively affect the accuracy of inferred data. However, several air quality monitors, including the ones we use in this work, perform an internal preprocessing of the raw data before sending them to the user or application. Preprocessing usually consists in smoothing the values of consecutive readings, in order to correct values affected by high level of noise. Since smoothing is already performed internally by the air quality monitor, in this work we perform a limited data cleaning, which consists of disregarding those portions of consecutive data where more than 50% of values are missing due to network errors or power failures.

Feature engineering

In order to reliably recognize food preparation activities, it is necessary to provide the machine learning algorithm with features useful to discriminate between cooking and non-cooking activities. For this reason, we have carefully analyzed the trend of air quality data when cooking was performed or not.

Figure 3 shows a screenshot of our air quality Web dashboard. The plot depicts the trend of carbon dioxide hourly average during a day. In that day, breakfast and lunch were prepared at around 7:30 a.m. and at around 1:30 p.m., respectively. From the plot, it is easy to observe that the absolute value of carbon dioxide is not sufficient to reliably distinguish cooking from non-cooking activities. Indeed, during all that day, the value of CO_2 was relatively stable, with a value slightly above 1000 ppm. The value of CO_2 started to increase in the morning at 7:00, when people went to the kitchen and initiated preparing breakfast. The increase of carbon dioxide levels was due both to the breathing of people in the kitchen and usage of a natural gas stove. The CO_2 value reached a local maximum at 9–10 a.m., and kept stable until 12:30 p.m., when a user opened the window to ventilate the kitchen. Soon after, a person started the preparation of lunch, and this activity determined an increase of carbon dioxide, whose value reached 1000 ppm and remained stable for the rest of the day.

From the analysis of Fig. 3, it emerges that, in order to distinguish cooking from non-cooking activities, it is important to analyse the trend of CO_2 levels, not only its absolute value. A similar point holds for the other parameters, such as the temperature, whose plot in the same day is reported in Fig. 4. For this reason, we engineered features taking into account not only the absolute values or averages, but also the difference between the current value and the past values. In particular, we build features considering the differences between the most recent value and the one in the previous 5, 10, 15, 20, and 25 min. We also use statistical features considering the average, minimum, and maximum value in the last 5 min, as well as the standard deviation of those values. Using these features, which are built using only the current values and past values, it is possible to recognize the current activity online; hence, we name this feature engineering modality *online feature extraction*.

However, especially to reliably determine the end of a cooking activity, it would be useful to observe the sensor data even after the end of the cooking activity. Indeed, the end of a cooking activity is often characterized by a drop of certain parameters, such as temperature, CO_2, and particulate matter; hence, the difference between those values during and after the cooking activity might generate characteristic spikes that are easy to recognize. Obviously, the use of features computed considering succeeding values determines a delay in the recognition process. Hence, for those applications having real-time requirements, such as the one addressed in this work, we only use features considering

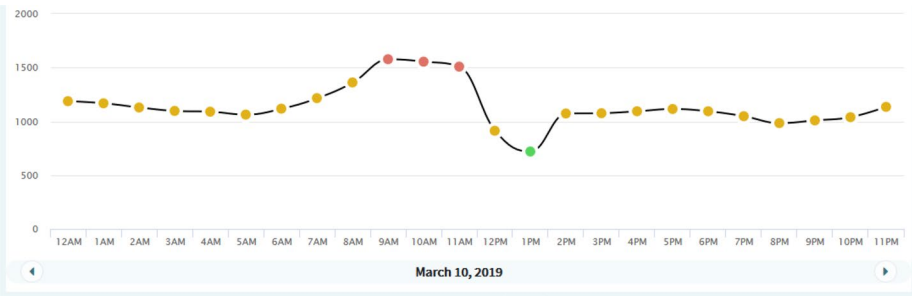

Fig. 3 Hourly trend of carbon dioxide in the kitchen in a day. Each point represents the average carbon dioxide value in the kitchen during a given hour. The points can take on different colors: green represents comfortable values for human life; red represents uncomfortable values; yellow represents intermediate values

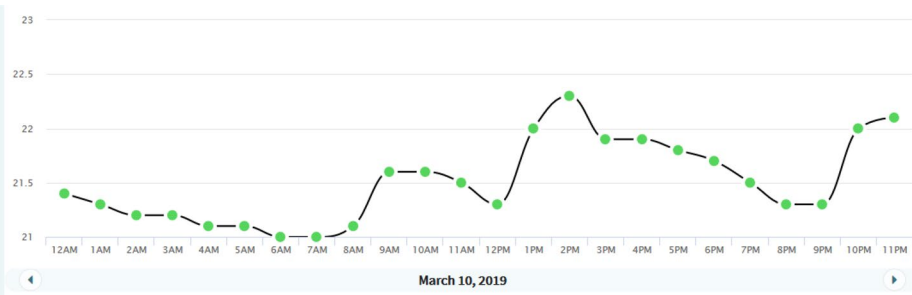

Fig. 4 Hourly trend of temperature in the kitchen in a day. Each point represents the average temperature value in the kitchen during a given hour. Green points represent comfortable values for human life

current and past values. For all the other applications, we also build features considering succeeding values in a temporal sliding window of 25 min. For instance, in order to build the feature vector referring to the activity executed at 12:00, we need to wait until 12:25, since the feature vector is built based on data acquired from 11:35 to 12:25. We name this feature engineering modality *delayed feature extraction*. In our experiments, reported in "Experimental evaluation" section, we evaluate both modalities.

It is worth to note that a single parameter is not sufficient to reliably recognize cooking activities. In general, increasing levels of carbon dioxide indicates the preparation of food using a gas stove; however, there may be some false positive when several people are in the kitchen, especially if the window is closed and the kitchen is small or poorly ventilated. Relying on CO_2 only, false negatives may happen when a meal is prepared without using a gas stove. For instance, in the day concerning Figs. 3 and 4, a dinner was prepared at around 9 p.m. using an electric oven. We can observe that the increase of CO_2 in that period of time was very limited, and due only to the sporadic presence of one person in the kitchen. The usage of the oven was clearly captured by the increase of the temperature. However, temperature alone is not a reliable parameter for food preparation, since it is strongly influenced by climatic factors and other external conditions. For this reason, we build our feature vectors considering six sensor data parameters: temperature, humidity, carbon dioxide, volatile organic compounds, particulate matter, and nitrogen dioxide. We disregarded carbon monoxide, ozone, and air pressure, because we experimentally found that they were not reliable indicators of food cooking.

Finally, the time of the day is another important indicator of food preparation, since cooking is normally carried out at specific times. We compute the current time of the day as the number of minutes that passed from midnight.

A deep neural network for food preparation

We have built a deep neural network for the classification task. The type of deep neural network chosen is a Multilayer Perceptron (MLP). Figure 5 shows its architecture. In particular, our MLP is composed by four layers: one input layer, two hidden layers, and

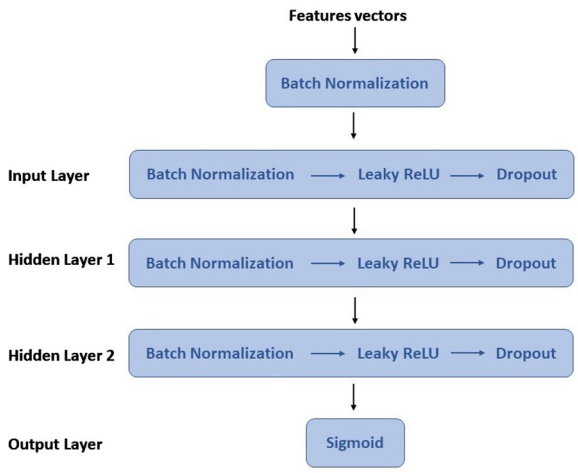

Fig. 5 Architecture of the proposed deep neural network

one output layer. The layers are fully connected (dense). The units per layer have been selected considering the number of features. Let nF be the number of features in input. The input layer has $nF/2$ number of units, the first hidden layer has exactly nF number of units, the second hidden layer has $2nF$ number of units. The output layer has only one unit, since we are performing a binary classification. We have used the Leaky Rectified Linear Units function (LeakyReLU) as activation function, with negative slope coefficient set at 0.2. Instead, for the output layer, we chose as activation the Sigmoid function, since we need a binary output value.

To prevent over-fitting we have added Dropout layers after every LeakyReLU layer. The fraction of the input units to drop has been set at 0.5. To speed up learning and to increase the stability of the neural network, we have also added Batch Normalization layers: one before the input layer and the other before every LeakyReLU layer. Batch Normalization layers have allowed us using a low learning rate (set at 0.0001) with Adam chosen as optimizer. As loss function, we used the binary cross-entropy function.

The deep neural network we have employed in this paper has been developed in Python programming language using the Keras framework[12] and Scikit-Learn library.[13] The used environment has been Google Colaboratory.[14]

The collected sensor data, the related annotations provided by humans, and the code related to the deep neural network we have developed can be freely downloaded from a GitHub repository.[15]

Experimental evaluation

In this section, we report the results of experiments carried out with an extensive set of real-world data.

Dataset

The dataset is composed of 350,551 data readings taken at each minute during more than 8 months in total from volunteers living in 8 different homes. The participants self-annotated the start and end time of cooking activities on a printed form, also specifying the kind of food that they cooked. At the end of data acquisition, the annotations were digitized by researchers using a custom program. The researchers actively interacted with the participants to clarify the meaning of those annotations that were ambiguous or unclear. The dataset was acquired in real-world environments and in naturalistic conditions; we did not rely on multiple annotators and we could not evaluate inter-rater reliability. As a consequence, even though the participants took annotation with care, the self-annotations inevitably may contain missing or wrong labels [35]. Data have been collected in homes having different characteristics, and in different periods of the year, to guarantee diversity and to ensure that the data represented real situations and conditions. In particular, six homes were situated in a city area by the sea with a Mediterranean climate, one in a big continental city, and one home was situated in a mountain

[12] https://keras.io/.

[13] https://scikit-learn.org/stable/.

[14] https://colab.research.google.com.

[15] https://github.com/FG2511/MLP_ForFoodPreparation.

area with alpine climate. Climate influences temperature and humidity, and the area (city vs. countryside) may influence air pollutants, particulate matter, and volatile organic compounds. In five homes, data was affected by the presence of people in the kitchen after the completion of the cooking activity, while in the other homes the meals were consumed in a different room. The season influences the frequency of other activities, such as opening the window or turning on a heating system, that may affect ambient and air parameters. The participants' age ranged from 23 to 71 years, with 10 females and 8 males. The volunteers were recruited among the families and mates of the authors, in order to include different kind of inhabitants. Specifically, homes included six different typologies of inhabitants: middle-aged single inhabitants, couples, families with children, groups of roommate students, a senior living alone, and a senior living with a middle-aged person. The participants did not receive any compensation for taking part to the study. All voluntaries were informed about the procedure used for data acquisition, the kind of data that would be acquired, the frequency of acquisition, and the kind of sensitive information that could be extracted from the acquired data. We explained that the data could be released in anonymous form to third parties for research purposes. In particular, we explained that we would not release any micro-data to third parties. Instead, we would release only aggregated macro-data; i.e., statistical feature vectors to be used for classification. Released data would not include neither explicit identifiers, nor quasi-identifier information. We also explained the potential impact of the research for supporting several kinds of medical conditions. The voluntaries gave written informed consent to their participation to the experiments. Each data record contains:

1. date,
2. time,
3. temperature (in °C),
4. relative humidity (in percentage %),
5. PM2.5 (Fine Particulate Matter in $\mu g/m^3$),
6. TVOC (Total Volatile Organic Compound in ppb),
7. CO_2 (Carbon Dioxide in ppm),
8. *CO* (Carbon Monoxide in ppm),
9. air pressure (in hPa),
10. O_3 (Ozone in ppb),
11. NO_2 (Nitrogen Dioxide in ppb),
12. current activity,
13. type of cooked food (e.g., rice, salad).

The *current activity* attribute can take two values: 1 if the user is cooking a meal, 0 otherwise. The number of data records with activity set to 1 is 16,323, while the remaining 334,228 are set to 0, meaning that "cooking" and "not cooking" classes are strongly unbalanced. As an example, Table 1 indicates a few records of the dataset with the information above.

The data are subject to many variables: the kind of person who cooked most in the house (3 men and 5 women, ages ranging from 20 to 72), the sensor distance from domestic appliances used for cooking (ranging from 5 cm to 1.5 m), the presence of

Table 1 Few examples of dataset records

Timestamp	Temp.	Hum.	PM2.5	TVOC	CO_2	CO	Pres.	O_3	NO_2
2018-11-24 13:13	27.7	60.14	5.54	66.0	442.0	0.0	1012.91	9.15	28.60
2018-11-24 13:14	27.7	60.21	4.56	67.0	461.0	0.0	1012.92	9.14	28.76
2018-11-24 13:15	27.7	59.84	8.37	67.0	465.0	0.0	1012.89	9.35	32.29
2018-11-24 13:16	27.6	58.96	6.19	67.0	467.0	0.0	1012.91	9.57	36.25

Each record is annotated with the current activity (1 if the user is cooking a meal, 0 otherwise), and the list of cooked food

air conditioning, pellet stove or windows in the kitchen, and the house structure (separate kitchen from the dining room or open-space). The data were acquired in different seasons; this feature strongly affects some parameters such as temperature and humidity.

Eight volunteers were given one air quality monitor to place in their homes, and for 1 month each they collected sensor data by writing down specific information each time they cooked: date, start and end time of cooking, cooked foods, domestic appliances used for cooking, and presence of an open window. The composition of home inhabitants was disparate, and included couples, students sharing a house, elderly living alone, and families with children.

Experimental setup

In order to optimize the deep neural network, it has been necessary to perform preliminary experiments to fine-tune the model parameters: activation function, optimizer, learning rate, batch size, dropout rate value. The optimized module that we devised is the one described in "A deep neural network for food preparation" section. As the classes in the dataset are strongly unbalanced (as explained in "Dataset" section), the class weights have been set up before the model generation.

Hence, we carried out several experiments, using two different types of validation.

1. Initially, the model has been evaluated using a one-shot split of the dataset. The model took 80% of the dataset as training set, the following 10% as validation set, and the remaining 10% for testing. The number of epochs has been decided using the Early Stopping function, which stops the evaluation when the loss function starts to increase. The patience parameter (i.e., the number of epochs with no improvement after which training stops) has been set to 2.
2. With the second type of validation, the model has been tested using a tenfold cross-validation. Specifically, we have used the Scikit-Learn KFold function to split the dataset. The split has been done maintaining the temporal order of the dataset by setting the shuffle parameter to False. This peculiarity is important, since shuffling the instances can introduce bias. Indeed, two instances that are contiguous in the dataset (i.e., two set of data measured at 1-min distance) are very similar: if an instance goes to the training set and the following goes to the test set, we have a bias.

We evaluate our model using two modalities. The first modality is named "minute-by minute", and considers each prediction, referring to a 1-min data, in isolation. The

second modality is named "cooking instance" recognition, and refers to the recognition of whole instances of cooking, where a cooking instance is a continuous interval of time during which the cooking activity took place.

For minute-by-minute modality, a correct recognition of *cooking* at minute m is counted as a true positive (TP). A false positive (FP) happens when the network predicts *cooking* at m, but cooking was not taking place at that minute. A false negative (FN) is counted if cooking was occurring at m, but the reasoner wrongly predicts "not cooking" for that minute. Finally, a true negative happens when the reasoner correctly predicts that cooking is not taking place at m.

For cooking instance modality, we consider each segment of contiguous minute-by-minute predictions of "cooking" starting at minute m and ending at minute n as the prediction of a single instance of cooking. Then, we count a TP if an actual instance of cooking has an intersection with a predicted cooking instance. If it has no intersection, then we count it as a FN. A FP occurs when (i) an actual instance of "not cooking" contains a predicted cooking instance, or (ii) a predicted instance of "cooking" contains an actual instance of "not cooking". A TN occurs when a predicted instance of "not cooking" does not contain an actual "cooking" instance.

The metrics used to evaluate the model are: accuracy, precision, recall, and F_1 score:

- accuracy is the percentage of correct predictions of the classifier and is defined as $\frac{TP + TN}{TP + TN + FP + FN}$;
- specificity $= \frac{TN}{TN + FP}$;
- precision $= \frac{TP}{TP + FP}$;
- recall $= \frac{TP}{TP + FN}$;
- F_1 score is the harmonic mean of precision and recall and is defined as $\frac{2 \cdot TP}{2 \cdot TP + FP + FN}$.

The results obtained have been improved with a post-processing step. The post-processing has been developed using a simple sliding windows algorithm. We have used More Itertools[16] library to implement sliding windows. The length of the windows has been set to 35 min. In each window, we look at the class of the central element (e.g., class 1), then we count the elements belonging to the same class. If they are less than a certain threshold (set at the half of the window plus one), the central element is set to the other class (e.g., class 0). In the other case, the class of the central element remains the same. The purpose of this step has been to remove small clusters of outliers and to merge close clusters of the same class.

Results

In the following, we present the experimental results. In all experiments, we applied tenfold cross-validation. Since air quality data values change relatively slowly with time, we have built the folds sequentially, in order to avoid the risk of having

[16] https://more-itertools.readthedocs.io/en/latest/.

Table 2 Results with minute by minute modality, online recognition

	R.F.	B.N.	N.B.	L.R.	*k*NN	SVM
TN	332,208	297,465	308,529	332,056	321,735	293,351
FP	1539	36,282	25,218	1691	12,012	40,396
TP	3691	11,403	7519	2665	4707	10,549
FN	12,626	4914	8798	13,652	11,610	5768
Accuracy (%)	*95.95*	88.23	90.28	95.62	93.25	86.81
Specificity (%)	*99.54*	89.13	92.44	99.49	96.40	87.90
Recall (%)	22.62	*69.88*	46.08	16.33	28.85	64.65
Precision (%)	*70.57*	23.91	22.97	61.18	28.15	20.71
F1-score (%)	34.26	*35.63*	30.66	25.78	28.50	31.37

Classifiers: Random forest (denoted as R.F., max depth = 10, iterations = 10), Bayes networks (B.N., using K2 hill climbing search algorithm), Naive Bayes (N.B.), *k* nearest neighbor (*k*NN, using $k = 1$), Logistic regression (L.R.), Support Vector Machines (SVM, using polynomial kernel and class balancing)

Numbers in italic indicate the largest value obtained in the experiment for each considered metric

consecutive data instances (which may be very similar among them, if not even identical) appearing in the training and test set, which could bias our results.

At first, we evaluated the performance of classification using different state-of-the-art machine learning algorithms. In these experiments, we used the Weka toolkit [13] for machine learning. Results obtained with minute-by-minute modality are shown in Table 2. Results show that the classification problem we are addressing is particularly challenging. Overall, among the evaluated classifiers, the one achieving highest accuracy was Random forest (95.95% accuracy). However, it is well known that, especially when classes are unbalanced, accuracy alone is not an adequate metric to evaluate the effectiveness of classification. Indeed, that classifier obtains good precision (70.57%), but low recall (22.62%), meaning that the predictions of "cooking" were quite reliable, but most cooking instances were actually not recognized. The classifier obtaining the highest F_1 score (35.63%) was Bayes networks, that (contrary to Random forest) exhibited good recall (69.88%), but low precision (23.91%). The Naive Bayes and Logistic regression classifiers obtained lower recognition rates than the former algorithms. The *k*NN classifier achieved a very good balance between precision (28.15%) and recall (28.85%); however, its overall recognition performance was low (F_1 score = 28.5%). The Support Vector Machines classifier obtained one the highest scores for recall (64.65%), but the lowest score for precision (20.71%), reaching an F_1 score of 31.37%.

Then, we performed classification using our deep learning model. Table 3 summarizes the results of online recognition obtained with minute-by-minute modality. Before post processing, despite the overall accuracy obtained being 87.95%, the overall F_1 score was slightly higher than 36%. Hence, the overall accuracy was comparable to the one obtained by the Bayesian network classifier, which was the one achieving the highest F_1 score in our pool of classifiers. However, our neural network obtained higher recall (73.85% vs.. 69.88%) and essentially the same precision (24.09% vs. 23.91%). Moreover, the size of the dataset was relatively small for training a deep neural network. We expect that the results of our deep neural network may significantly increase using additional training data. For these reasons, we decided to use the deep

Table 3 Deep neural network

	Original predictions	After post-processing
TN	296,219	301,258
FP	37,978	32,939
TP	12,055	11,985
FN	4269	4339
Accuracy (%)	87.95	*89.36*
Specificity (%)	88.64	*90.14*
Recall (%)	*73.85*	73.42
Precision (%)	24.09	*26.68*
F_1 score (%)	36.33	*39.14*

Results with minute by minute modality, online recognition

Numbers in italic indicate the largest value obtained in the experiment for each considered metric

Table 4 Deep neural network

	Original predictions	After post-processing
TN	29,3027	297,008
FP	41,140	37,159
TP	12,687	12,461
FN	3637	3863
Accuracy (%)	87.22	*88.30*
Specificity (%)	87.69	*88.88*
Recall (%)	*77.72*	76.34
Precision (%)	23.57	*25.11*
F_1 score (%)	36.17	*37.79*

Results with minute by minute modality, delayed recognition

Numbers in italic indicate the largest value obtained in the experiment for each considered metric

neural network in the rest of the experiments. The relatively low recognition rates that we achieved may be probably due to the fact that the dataset is strongly imbalanced, since time of cooking covers less than 5% of the dataset. For this reason, it was hard for the neural network to identify the few "cooking" activities within the vast majority of "not cooking" instances. Moreover, the dataset was acquired in several different real-world conditions. In particular, our neural network achieved good recall, but low precision. Results were slightly improved by post-processing, reaching an F_1 score close to 40%. By inspecting the results, we observed that post-processing improved the precision by around 3% without negatively impacting recall. We repeated the same experiments with delayed recognition. We recall from "Feature engineering" section that in this modality the recognition of the current activity is delayed by 25 min in order to consider the succeeding trend of air quality values. With minute-by-minute recognition, we observed that delayed recognition achieved essentially the same accuracy of online recognition, as shown in Table 4. We performed a statistical study in order to understand whether the difference in the results obtained with online vs. offline recognition was statistically significant. For this reason, we applied

the well-known measures of Φ coefficient and χ^2 test [12] to the output of the classifiers using the two recognition methods. We recall that the Φ value of two binary variables having identical distribution tends to 1, while the p value of χ^2 test tends to 0. In our case, we obtained a Φ value of 0.90, and the p value of the χ^2 test of 2.2e−16. Hence, we can conclude that the two techniques produced results that are statistically very similar for minute-by-minute classification.

However, for our application, it is important to identify whole instances of cooking activities, not the single minutes during which the activity takes place. In cooking instance modality, our online recognition method achieved better results than in minute-by-minute modality. Results can be found in Table 5. In particular, before post-processing, the technique achieved an F_1 score slightly lower than 46%. This modality significantly increased both precision (from 24.09 to 32.21%) and recall (from 73.85 to 78.77%). Results were further improved by post processing, reaching an overall F_1 score close to 60%. In particular, post-processing provided more balance between precision and recall values. Note that, after post-processing, the total number of predicted instances was strongly reduced, and this fact had obviously an impact on the overall numbers of TN, FP, TP, and FN. The reduction of the total number of predicted instances was due to the fact that our post-processing algorithm merged multiple predicted cooking instances that were temporally close. The reader is referred to "Experimental setup" section for the definition of cooking instance modality. In cooking instance modality, accuracy improved using delayed recognition (Table 6), achieving an F_1 score larger than 62%.

Discussion

Overall, despite we carefully designed the deep neural network, the achieved results are not fully satisfactory. This fact may be explained in several ways.

- First of all, while the dataset includes both hot and cold meals, our system is suited to recognize only the former. Indeed, it fails to recognize the majority of cold meals. This is an intrinsic limitation of any recognition system based on air quality data. In order to recognize cold meals, different kinds of sensors should be added to the system.

Table 5 Deep neural network

	Original predictions	After post-processing
TN	4506	727
FP	1109	482
TP	527	491
FN	142	178
Accuracy (%)	*80.09*	64.86
Specificity (%)	*80.25*	60.13
Recall (%)	*78.77*	73.39
Precision (%)	32.21	*50.46*
F_1 score (%)	45.73	*59.81*

Results with cooking instance modality, online recognition

Numbers in italic indicate the largest value obtained in the experiment for each considered metric

Table 6 Deep neural network

	Original predictions	After post-processing
TN	3988	698
FP	957	454
TP	549	505
FN	120	164
Accuracy (%)	*80.82*	66.06
Specificity (%)	*80.65*	60.59
Recall (%)	*82.06*	75.49
Precision (%)	36.45	*52.66*
F_1 score (%)	50.48	*62.04*

Results with cooking instance modality, delayed recognition

Numbers in italic indicate the largest value obtained in the experiment for each considered metric

- Secondly, the dataset was acquired in disparate real-world conditions. Homes included single inhabitants, couples, families with children, and groups of roommate students. Of course, the age and number of inhabitants has an impact on the kind and quantity of cooked food, and consequently on the change in air quality conditions determined by cooking. The topology of the home also has an impact on air quality data. Indeed, if inhabitants consume the meal within the kitchen, their presence determines an increase of temperature and CO_2 levels even after cooking has ended. If the inhabitants consume the meal in a different room, the CO_2 and temperature levels in the kitchen decrease as soon as cooking is finished. In our dataset we had both cases, depending on the home. This aspect could be taken into account by selecting only the subset of the training data acquired in conditions that resemble those of the target environment.
- Thirdly, being manually annotated, the dataset labels have an inevitable level of noise, which may include wrong start and end time of cooking execution, or wrong labels.

Nonetheless, considering that each activity recognition system has a considerable error rate, our system based on air quality data can be coupled with other activity recognition tools in the home to increase the overall activity recognition rate. For instance, the accuracy of the system may be increased by coupling our air-quality based system with other sensors attached to kitchen furniture and instruments. Moreover, as explained in "Use case on a robotic platform" section, the user's feedback resulting from the interaction with the robot is used to periodically re-train the neural network using additional training data. Hence, we expect the accuracy of the system to increase with time thanks to human–robot interaction. Even though we did not evaluate this aspect in our experiments, we also believe that the number of false positives may be significantly reduced thanks to the usage of computer vision APIs of the robot, as described in "Architecture of the use case" section.

The average execution time of the neural network algorithm for recognizing an instance of data is 0.0317 ms on a cloud computing infrastructure. Hence, our system is feasible for real-time applications as in the proposed use case.

Use case on a robotic platform

In this section we are going to describe the use case we have set within the social robotics domain. A humanoid robot has been employed to interact with the user when the system recognizes that something is being cooked. In such a case, the robot asks the user what he/she is cooking. More in detail, "Zora, the used humanoid robot" section will include details of the robotic platform we have adopted whereas "Architecture of the use case" section will include the architecture of the use case we have designed.

Zora, the used humanoid robot

The Zora robot[17] uses the same robotic infrastructure of Nao, an autonomous, programmable humanoid robot developed by Aldebaran Robotics, a French robotics company headquartered in Paris, which was acquired by SoftBank Group in 2015 and re-branded as SoftBank Robotics. With respect to Nao, Zora adds an extremely simple and intuitive user interface that allows any user to play loaded behaviours (apps, dances and games targeting care, kids, STEM[18] market), to give action commands to the robot to change posture and move each part of her body, to make her talk in eight possible languages, and to use the Composer to create simple robot behaviors, composing a sequence of actions in a visual environment where no programming knowledge is needed.

Like Nao, Zora is also completely programmable through the Choregraphe suite,[19] which allows users to:

- create and combine different robot behaviours using a visual approach making use of the Python programming language;
- develop animations by leveraging an intuitive and dedicated user interface;
- test the robot behaviours and animations on either the simulated robot or the real one;
- develop complex behaviours and human–robot interactions by leveraging calls to REST APIs of external services on the Internet.

In order to capture the voice of the user when he/she speaks, the robot is equipped with four microphones, two of them in the front of the head and two at the back. The robot can therefore record the human voice, which is contextually analyzed and transformed into text by a speech recognition module powered by Nuance.[20] However, we are currently relying on cloud computing systems for speech recognition in order to improve the accuracy of the speech to text process. In fact, this allows us pre-processing the sound recorded by Zora and removing noise (e.g background noise, fan noise, etc.), which may compromise the conversion of human voice into written text. As such, the resulting audio file is sent to IBM Watson Speech to Text[21] to perform speech recognition. Figure 6 shows an image of Zora.

[17] https://www.youtube.com/watch?v=lO52sLF-u_4&t=1s.

[18] Science, technology, engineering, and mathematics.

[19] http://doc.aldebaran.com/1-14/software/choregraphe/index.html.

[20] https://www.nuance.com.

[21] https://www.ibm.com/watson/services/speech-to-text/.

Fig. 6 An image of Zora, the employed humanoid robotic platform

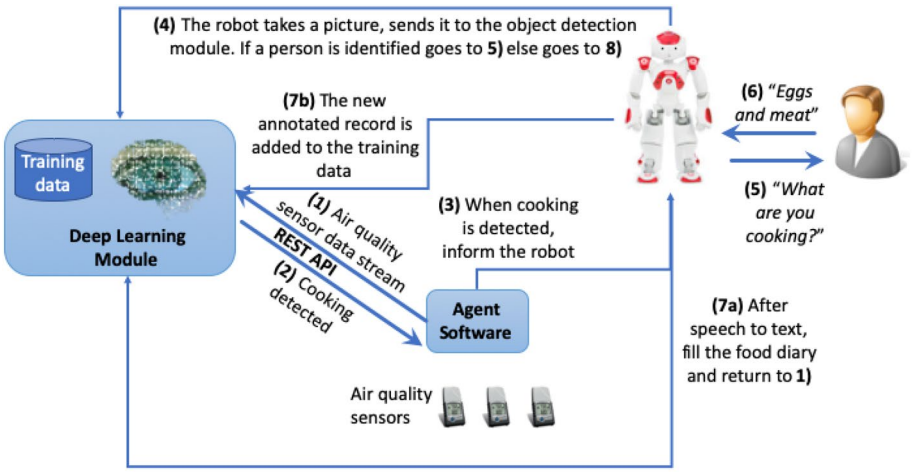

Fig. 7 Overall architecture of our use case prototype implementation. Note that point (4) is optional: the user may enable or not the camera-based object detection of the robot. If (4) is disabled, action goes directly from point (3) to point (5)

Architecture of the use case

Figure 7 shows the architecture of the proposed use case. A Deep Learning module contains the annotated data and the trained deep learning model. That module exposes REST APIs to classify as *cooking* or *non cooking* a new collected record of sensor data. One more software agent, periodically, collects sensor data and calls the REST APIs of the Deep Learning module. If the new read data is classified as *cooking* then this is communicated to the robot via socket communication. Before starting the interaction with the user, the robot checks if someone is actually in the kitchen. For such a purpose, it takes a picture of the environment, which is sent to an object detection module to identify potential persons. However, for the sake of privacy, the object detection task

is optional, and the user may decide whether to activate it or not. More specifically, we have employed the TensorFlow Object Detection API,[22] which provides an open source framework built on top of TensorFlow that makes it easy to construct, train and deploy object detection models. TensorFlow Object Detection API can be used with different pre-trained models. More in detail, we have chosen a Single Shot MultiBox Detector (SSD) model with Mobilenet (ssd_mobilenet_v1_coco) which had been trained using the Microsoft COCO dataset,[23] consisting of 2.5M labeled instances in 328000 images, containing 91 object types. The ssd_mobilenet_v1_coco is reported to have mean average precision (mAP) of 21 on the COCO dataset. For further details on the SSD and the evaluation carried out on the COCO dataset please check the work of authors in [19]. The back-end of the object detection module has been embedded into a server-side application which exposes REST APIs that, given an input image, return the bounding box of each recognized object in the image along with a category and a confidence value. We considered valid only the objects that were recognized with a confidence value equal or higher than 60%. The back-end is hosted within the Deep Learning Module.

When a cooking instance is recognized, the robot starts the interaction with the user. If camera-based recognition is enabled, the robot takes six pictures in the kitchen, each 60 degrees distant from the other. If the robot identifies one or more persons (class *person* of the COCO dataset) in the images, it asks what food the user is preparing. If camera-based recognition is disabled, the robot makes its question in any case. Once the user replies, the robot performs speech to text processing and sends the extracted food as well as the sensors data to the Deep Learning module which extends its training data with the new annotation and, periodically, retrains the overall model. Note that, in the current implementation of our system, the speech interaction to acquire food journaling data is over-simplistic, being based on a simple question-answer paradigm. Since most food journaling applications require detailed information about the kind and quantity of food, we will investigate a more sophisticated conversational agent for food journaling in future work. Voice-based identification methods will also be used to recognize the inhabitant, in case of multi-resident homes. If the classifier forecasts a *cooking* activity using the current air quality data and the user is not actually cooking (i.e. the user replies *nothing* to the robot question above) the classification is wrong and the new record is sent to the Deep Learning Module together with the *not cooking* label. If the object detection module does not identify any person in the kitchen, it overrides the classification of the Deep Learning Module thus reducing the false positives and improving the overall classification. Moreover, the new pair (sensor data, *not cooking*) is sent as further training element. Periodically and when enough new annotated data have been collected, the Deep Learning Module trains again the model.

We would like to point out that the robot has not been employed for the collection of the annotated data during the 8 months from the annotators. During that period, only our air quality sensors were employed and their measurements were saved for the whole period. After the creation of our gold standard and the training of our model, we set up the whole architecture shown in Fig. 7 for a preliminary test on real

[22] https://bit.ly/2lPqHJk.

[23] http://cocodataset.org.

settings. Advanced methods to improve our use case, including techniques for optimized path planning of the mobile robot [30], will be investigated in future work.

Preliminary results on human–robot interaction mechanism

After having collected all the information related to the air quality sensors, in one of the houses of the annotators we performed a preliminary technical validation of the human–robot interaction mechanism. In order to interact with the user, the robot employs a state of the art object detection classifier, text-to-speech and speech-to-text technologies, which are widely evaluated in the literature. We have already mentioned the used speech-to-text technologies. As far as the object detection is concerned, we have used the classifier based on the work of authors in [18]. The object detection software (that was enabled in our use case) and the classification software (to identify a cooking instance out of the air sensor data) were run in a pc we brought in the house to perform the test together with the robot and the air sensors. The whole human–robot interaction architecture has been preliminarily tested for short time (one full day), and the only errors we noticed occurred because of the wrong prediction of the activity recognition module. As mentioned earlier in the paper, the human–robot interaction has been kept simple. To easily recognize the food spoken by the user, we first collected a list of food items online that were enriched by each of the annotators. Basically, we asked each of them to write the list of food items they have cooked or might cook in the future. After we removed the duplicates, we obtained a list of 86 food items that were organized in a two-levels hierarchical structure. The first layer contained general terms, whereas the second levels contained items that were associated to one food item of the first level. For example, *egg* is a general item, while *omelette* is a specific item related to *egg*. Therefore, when the machine learning module predicted a cooking activity (and the robot identified a person in the kitchen), the robot asked the user what he/she was cooking. Out of the natural language expressions spoken by the user, after the robot performed speech-to-text, it was just a matter of recognizing terms we had in the vocabulary without performing any comprehension of the semantics involved in the natural language text. This process did not lead to any errors and all the spoken items have been correctly identified within the defined vocabulary. There was one researcher present in the morning during the first cooking activity and in the evening during the last cooking activity that monitored the human robot interaction after having trained the English speaking person living in the house and informed her on the behaviour of the robot. Some facts, comments and impressions that turned out from our preliminary experiment were the following:

- the object detection module correctly identified all the times when someone was in the kitchen;
- one out of five cooking activities was not recognized as a cooking activity by the classifier;
- two times the robot thought that there was a cooking activity (two false positives occurred of the Deep Learning module): that was fixed as soon as the user replied

Table 7 Five entries of the food diary filled during 1 day through the human robot interaction use case

Time	Food
8:11	Coffee
13:18	Pasta and potatoes
17:09	Chocolate
20:23	Broccoli and steak
22:34	Tea

nothing to the robot question *What are you cooking?* and the correct pair (sensor data, no cooking activity) was sent to the Deep Learning module;

- we have developed everything (vocabulary, human robot interaction, etc.) in English and the user involved within our experiment was an English speaker;
- out of five cooking activities, there were not cases when the user mentioned a food not present within the dictionary we had prepared;
- the entries we filled in our food diary for the short experiment are depicted in Table 7;
- we asked what the impressions of the user interacting with the robot were and she was very curious and excited to talk it. She did not think the robot was intrusive and liked the simple human robot interaction we designed. She would even have liked if the robot could have entertained her with songs, music, radio, or simple interaction or question-answering capabilities provided, for example, by voice assistant tools today.

Conclusion and future work

In this paper, we have laid the foundation of a novel method to support food journaling, addressed to frail people living alone. Our system relies on advanced air quality sensors for cooking recognition. We have shown the process of collecting and analysing air quality sensor data to detect when the user is cooking in order to trigger the interaction with a digital agent to acquire food data. We have developed a deep neural network trained on a large dataset acquired during 8 months in disparate conditions by different people. An experimental evaluation has been carried out to assess the accuracy of the model on the given classification problem and the feasibility of the method for real-time applications. We have also developed an initial prototype considering a use case where a social robot interacts with the neural network and with the user. Our preliminary prototype is the first in its kind and shows several challenges we had to face and many more that we still need to address. However, we believe that the technologies to address these challenges are out there and our work provides a significant step in this direction.

Several challenges to be addressed in future work remain open. First of all, we will investigate methods to increase the accuracy of our cooking recognition system. An obvious direction is to couple the air quality sensor with other sensors to recognize the preparation of cold meals. As explained in "Feature engineering" section, the temperature alone is not sufficient to reliably recognize cooking activities, because indoor

temperature is influenced by external temperature. A similar point holds for humidity and other factors. In order to mitigate the influence of external conditions, we could include additional data taken from online weather services. Since both the topology of the home and the characteristics of inhabitants (including their number and age distribution) affect the air quality conditions at cooking time, we will investigate techniques to couple our data-driven method with a knowledge-based one, to fine-tune recognition to home's and inhabitants' characteristics. Other domain knowledge, such as the expected duration of cooking activities, may be used to improve the recognition rates of our cooking recognition system, and this is a research direction we will pursue. Future work also includes the definition of an effective and engaging conversational interface for interactively filling the food diary, and voice-based identification methods to recognize the current inhabitant in case of multi-resident homes. For such a purpose, one direction we are heading is to employ Google Assistant technology for the human–robot interaction exploiting the APIs and open source tools (e.g. DialogFlow) that Google makes available to the community. We would like to extend the vocabulary we have defined according to Semantic Web best practices in order to have a more comprehensive ontology involving all the food items that might be cooked according to any international recipes. Finally, we plan to execute extensive tests and a much more comprehensive evaluation of the human robot interaction approach based on our preliminary use case prototype implementation.

Acknowledgements
The authors would like to thank the anonymous reviewers for their insightful comments and suggestions.

Authors' contributions
All authors gave equal contribution. All authors read and approved the final manuscript.

Funding
This work was supported by the Open Access Publishing Fund of the University of Cagliari, with the funding of the Regione Autonoma della Sardegna - L.R. n. 7/2007.

Availability of data and materials
Data and source code used in our experiments are available online (https://github.com/FG2511/MLP_ForFoodPreparation).

Competing interests
The authors declare that they have no competing interests.

Received: 19 July 2019 Accepted: 25 May 2020
Published online: 17 June 2020

References

1. Abdullahi L, Delgado-Saborit JM, Harrison R (2013) Emissions and indoor concentrations of particulate matter and its specific chemical components from cooking: a review. Atmos Environ 71:260–294. https://doi.org/10.1016/j.atmosenv.2013.01.061
2. Achananuparp P, Lim E, Abhishek V (2018) Does journaling encourage healthier choices? Analyzing healthy eating behaviors of food journalers. In: Kostkova P. Grasso F, Castillo C, Mejova Y, Bosman A, Edelstein M (eds) Proceedings of the 2018 international conference on digital health, ACM, pp 35–44
3. Amft O, Stäger M, Lukowicz P, Tröster G (2005) Analysis of chewing sounds for dietary monitoring. In: UbiComp 2005: ubiquitous computing, 7th international conference, Lecture Notes in Computer Science, vol 3660, Springer, Berlin, pp 56–72
4. Bouwman L, Hiddink GJ, Koelen MA, Korthals M, van't Veer P, van Woerkum C, Personalized nutrition communication through ict application (2005) Personalized nutrition communication through ict application: how to overcome the gap between potential effectiveness and reality. Eur J Clin Nutr 59:108–116
5. Brunoand V, Resende S, Juan C (2017) A survey on automated food monitoring and dietary management systems. J Health Med Inform 8(3):1–15
6. Casas J, Mugellini E, Khaled OA (2018) Food diary coaching chatbot. In: Proceedings of the 2018 ACM international joint conference and 2018 international symposium on pervasive and ubiquitous computing and wearable computers, ACM, pp 1676–1680

7. Celino I, Calegari GR (2020) Submitting surveys via a conversational interface: an evaluation of user acceptance and approach effectiveness. Int J Hum Comput Stud 139:1–16
8. Chi P, Chen J, Chu H, Lo J (2008) Enabling calorie-aware cooking in a smart kitchen. In: PERSUASIVE, Lecture Notes in Computer Science, vol 5033, Springer, Berlin, pp 116–127
9. Cordeiro F, Bales E, Cherry E, Fogarty J (2015) Rethinking the mobile food journal: Exploring opportunities for lightweight photo-based capture. In: Proceedings of the 33rd annual ACM conference on human factors in computing systems (CHI), ACM, pp 3207–3216
10. Cordeiro F, Epstein DA, Thomaz E, Bales E, Jagannathan AK, Abowd GD, Fogarty J (2015) Barriers and negative nudges: Exploring challenges in food journaling. In: Proceedings of the 33rd annual ACM conference on human factors in computing systems (CHI 2015), ACM, pp 1159–1162
11. DiFilippo KN, Huang WH, Andrade JE, Chapman-Novakofski KM (2015) The use of mobile apps to improve nutrition outcomes: a systematic literature review. J Telemed Telecare 21(5):243–253
12. Guilford JP (1941) The phi coefficient and chi square as indices of item validity. Psychometrika 6(1):11–19
13. Hall MA, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. SIGKDD Explor 11(1):10–18
14. Hameed I (2016) Using natural language processing (nlp) for designing socially intelligent robots. In: Conference: 2016 joint IEEE international conference on development and learning and epigenetic robotics (ICDL-EpiRob), pp 268–269. https://doi.org/10.1109/DEVLRN.2016.7846830
15. Hollis JF, Gullion CM, Stevens VJ, Brantley PJ, Appel LJ, Ard JD, Champagne CM, Dalcin A, Erlinger TP, Funk K, Laferriere D, Lin PH, Loria CM, Samuel-Hodge C, Vollmer WM, Svetkey LP (2008) Weight loss during the intensive intervention phase of the weight-loss maintenance trial. Am J Prev Med 35:118–126
16. Kazakos E, Nagrani A, Zisserman A, Damen D (2019) Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In: 2019 IEEE/CVF international conference on computer vision, IEEE, New York, pp 5491–5500. https://doi.org/10.1109/ICCV.2019.00559.
17. Krebs-Smith SM, Pannucci TE, Subar AF, Kirkpatrick SI, Lerman JL, Tooze JA, Wilson MM, Reedy J (2018) Update of the healthy eating index: Hei-2015. J Acad Nutr Diet 118(9):1591–1602
18. Liu S, Qi L, Qin H, Shi J, Jia J (2018) Path aggregation network for instance segmentation. In: 2018 IEEE/CVF conference on computer vision and pattern recognition, pp 8759–8768
19. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: single shot multibox detector. In: Proceedings of the European conference on computer vision (ECCV) (2016). arXiv:1512.02325
20. Lukoff K, Li T, Zhuang Y, Lim BY (2018) Tablechat: mobile food journaling to facilitate family support for healthy eating. Proc ACM Hum Comput Interact 2:114:1–114:28
21. Mamykina L, Mynatt ED, Kaufman DR (2006) Investigating health management practices of individuals with diabetes. In: Proceedings of the 2006 conference on human factors in computing systems (CHI), ACM, pp 927–936
22. Mankoff J, Hsieh G, Hung HC, Lee S, Nitao E (2002) Using low-cost sensing to support nutritional awareness. In: UbiComp 2002: ubiquitous computing, 4th international conference, Lecture Notes in Computer Science, vol 2498, Springer, Berlin, pp 371–376
23. Marr JW (1971) Individual dietary surveys: purposes and methods. World Rev Nutr Diet 13:105–164
24. Matuszek C (2018) Grounded language learning: where robotics and nlp meet. Proc IJCAI 2018:5687–5691. https://doi.org/10.24963/ijcai.2018/810
25. Michie S, West R, Sheals K, Godinho CA (2018) Evaluating the effectiveness of behavior change techniques in health-related behavior: a scoping review of methods used. Transl Behav Med 8(2):212–224
26. Oh H, Nguyen J, Soundararajan S, Jain R (2018) Multimodal food journaling. In: Boll S, Jain R, O'Connor NE, McDaniel TL, Meyer J (eds) Proceedings of the 3rd international workshop on multimedia for personal health and health care, ACM, pp 39–47
27. Riboni D (2019) Opportunistic pervasive computing: adaptive context recognition and interfaces. CCF Trans Pervasive Comput Interact 1(2):125–139
28. Rohrbach M, Amin S, Andriluka M, Schiele B (2012) A database for fine grained activity detection of cooking activities. In: IEEE conference on computer vision and pattern recognition, IEEE Computer Society, pp 1194–1201
29. Romano KA, Becker MAS, Colgary CD, Magnuson A (2018) Helpful or harmful? the comparative value of self-weighing and calorie counting versus intuitive eating on the eating disorder symptomology of college students. Eating Weight Disord Stud Anorexia Bulimia Obes 23(6):841–848
30. Saeed RA, Recupero DR, Remagnino P (2020) A boundary node method for path planning of mobile robots. Robot Auton Syst 123:103320
31. Sen S, Subbaraju V, Misra A, Balan RK, Lee Y (2018) Annapurna: building a real-world smartwatch-based automated food journal. In: 19th IEEE international symposium on "A World of Wireless, Mobile and Multimedia Networks", IEEE Computer Society, pp 1–6
32. Wang G, Cheng S, Lang JL, Wen W, Wang X, Yao S (2016) Characterization of volatile organic compounds from different cooking emissions. Atmos Environ 145 https://doi.org/10.1016/j.atmosenv.2016.09.037
33. Wilde MH, Garvin S (2007) A concept analysis of self-monitoring. J Adv Nurs 58:339–350
34. Wildenbos GA, Peute LWP, Jaspers MWM (2018) Aging barriers influencing mobile health usability for older adults: a literature based framework (MOLD-US). Int J Med Inform 114:66–75
35. Woznowski P, Tonkin E, Laskowski P, Twomey N, Yordanova K, Burrows A (2017) Talk, text or tag? The development of a self-annotation app for activity recognition in smart environments. In: IEEE international conference on pervasive computing and communications workshops, IEEE, New York, pp 123–128
36. Yordanova K, Lüdtke S, Whitehouse S, Krüger F, Paiement A, Mirmehdi M, Craddock I, Kirste T (2019) Analysing cooking behaviour in home settings: towards health monitoring. Sensors 19(3):646
37. Yordanova K, Whitehouse S, Paiement A, Mirmehdi M, Kirste T, Craddock I (2017) What's cooking and why? behaviour recognition during unscripted cooking tasks for health monitoring. In: IEEE international conference on pervasive computing and communications workshops, IEEE, New York, pp 18–21

38. Zhu F, Bosch M, Woo I, Kim S, Boushey CJ, Ebert DS, Delp EJ (2010) The use of mobile devices in aiding dietary assessment and evaluation. J Sel Topics Signal Process 4(4):756–766
39. Zhu N, Diethe T, Camplani M, Tao L, Burrows A, Twomey N, Kaleshi D, Mirmehdi M, Flach PA, Craddock I (2015) Bridging e-health and the internet of things: the SPHERE project. IEEE Intell Syst 30(4):39–46

Publisher's Note