Smart Food Scanner System Based on Mobile Edge Computing

Bahman Javadi, Quoc Lap Trieu, Kenan M. Matawie, Rodrigo N. Calheiros
School of Computer, Data and Mathematical Sciences
Western Sydney University, Australia
Emails: {b.javadi, k.matawie, r.calheiros}@westernsydney.edu.au, 19263045@student.westernsydney.edu.au

***Abstract*—Smart applications, including Internet of Things (IoT) and Big Data analytics, are traditionally hosted by cloud infrastructures, which can result in high latency and cost beyond users expectation. Edge computing has emerged as a paradigm that can alleviate the pressure on clouds by delegating parts of the computation to devices in the edge of the network, at closer proximity to end users and IoT devices. In this paper, we discuss a smart application, built on top of mobile edge computing concept, to enables users to measure and analyse their food intake and support nutritional decision-making. The approach utilizes mobile edge computing to offload application computations and communications to the edge, thus saving battery life, increasing the processing capacity, and improving user comfort. In order to develop this system, we propose a loosely coupled architecture for a smart food scanner and then implement it using various IoT sensors. The performance evaluation results reveal that the implemented system can be used as an interactive appliance by users with minimum dependency and usage of their mobile phones.**

1. Introduction

The last decade witnessed the consistent growth of cloud computing as the platform of choice for existing and emerging distributed applications, ranging from multi-tier web applications hosting anything to CPU-intensive, complex scientific applications [1]. The success of cloud platforms are attributed to its capacity to support virtually infinite amount of resources for applications on demand, and the native support for highly reliable and elastic applications when solutions are appropriately architected. However, recent technology trends started to expose the limitations of clouds to host smart applications. Innovations in the area of Internet of Things (IoT) [2], Big Data [3], and deep learning [4] encouraged the development of smart applications that expose data ingestion as a major bottleneck of centralized cloud solutions.

The large volume of data generated by IoT sensors can put a considerable pressure on clouds and depending on the pricing model of cloud and Internet providers, result in large application costs. Conversely, deep learning models demand large datasets for accurate classification, which impacts data transfer (to move the datasets to the cloud) and computing costs.

Moreover, the fact that cloud application are hosted on large scale, centralized data centers that usually are located far from users, add another issue of latency-induced delays in response time, which affect user experience and may have adverse effects on the application success and profitability. For example, content delivery network provider Akamai estimates that a delay in 100 ms in the application response time result in a loss of revenue in a drop of conversion rate of 7% [5]. Thus, it is clear that new demands that arose since cloud computing emerged are not suitably addressed by clouds. Zhang et al. [6] makes a more comprehensive analysis of the limitations of cloud platforms, with emphasis on its support for IoT. They identified seven key issues with cloud platforms that have a large impact on IoT applications. Aligned with our earlier discussion, network latency and bandwidth are among those identified issues, and so are scalability and Quality of Service (QoS) guarantees. Recently, edge computing [7], in which computing and storage nodes are placed at the networks edge in close proximity to users, has grown dramatically. This technology promises to deliver responsive computing services, scalability, privacy enforcement and the ability to mask transient cloud outages.

In parallel to the advances in cloud computing, dramatic improvements in mobile computing transformed mobile phones to complete computing systems with enough processing power and memory resources to host smart applications. However, battery life is the main factor limiting smartphones to perform computing-intensive tasks. The field of Mobile cloud computing (MCC) [8] emerged to enable mobile devices to outsource computing tasks to the cloud. Other approaches involved exploring edge devices as the offloading destination of mobile computations—what became known as mobile edge computing (MEC) [9].

Because part of the computation (and related data communication) that would otherwise occur in the cloud is now occurring at the edge of the network in proximity to end-users (or IoT sensors), latency, and the overall network data transfer, can be reduced considerably with edge computing, which impacts on response time, Quality of Service, and data transfer costs. Furthermore, it also has an extra benefit of limiting the amount of data going to cloud providers, which can also have security and privacy benefit for users. One approach commonly used for edge computing is to outsource to the cloud heavy computation (as in general the edge is assumed to be resource-constrained) such as deep learning training (as in the work by Li et al. [10]) and other latency-insensitive tasks, whereas the remainder of the computation occurs in the edge.

In this paper, we discuss a smart application, built on top of mobile edge computing concept, to enables users to measure and analyse their food intake and support nutritional decision-making. The approach utilizes mobile edge computing to offload application computations and communications to the edge, thus saving battery life, increasing the processing capacity, and improving user comfort. This edge application was enabled by our initial findings when investigating a solution for non-invasive food intake detection [11]. After investigation on the particular interfaces, elements, and context-independent components in our early application, we were able to design the proposed application to leverage edge computing. In particular, the contributions of this paper are the following: (i) An edge-based computing system that is able to aggregate heterogeneous IoT devices and edge resources with powerful cloud resources in the backend to support non-invasive capture of nutritional information data; and (ii) an implementation and evaluation of the proposed system, which reveals the capabilities of edge computing in the area of smart applications.

The rest of the paper is organized as follows. Section 2 presents a comprehensive review of related work in edge computing frameworks and systems supporting nutrition information capture. Section 3 discusses the motivation for the system, its requirements and the proposed architectural and system implementation. Section 4 presents an evaluation of the implemented system and results of service time and power consumption. Conclusions and future works are discussed in Section 5.

2. Related Work

We organize our related work based on key themes presented in this paper, namely edge computing frameworks and current approaches for nutritional information capture.

2.1. Edge Computing Frameworks

A number of frameworks for edge computing have been proposed in the literature [12]–[17]. Such frameworks target some "general" model for Edge computing, without considering specific applications. Approaches vary regarding the nature of edge devices and the degree of ownership and control users or Cloud providers have over edge or IoT devices (data sources). Although some of them have target application areas providing guidance over architectural elements of the framework (such as medical cyber-physical systems [13] or vehicular networks [14]), in common across these approaches is also the fact that little attention is given to the applications that need to be supported; applications are heavily abstracted, making hard to map specific applications to the scenarios proposed.

Some frameworks recently proposed [18]–[20] target specifically mobile edge platforms, and they usually address the specific issue of offloading the computation from mobile devices to the edge or the cloud. Our work also contain elements that are relevant in this context, as some existing approaches for food detection were developed as mobile applications, whereas our approach perform the same work on the edge and the cloud.

As a result of such overgeneralization of applications by existing frameworks, existing Edge applications do not leverage them, and usually embed their own solution for integration of the diverse layers of the stack. Nevertheless, a few applications emerged in the area of health and well-being. The closest application to our approach is proposed by Liu et al. [21]. Such application consists of a deep learning-based food recognition system deployed on the edge. At the user side, a mobile app captures the photo of a food. The mobile phone itself is the Edge device, which carries out preprocessing and image segmentation. The system leverages the cloud for training a convolutional Neural Networks for food recognition and for classification of the photos segmented by the Edge device. Our approach focus not only on the food detection, but also on the estimation of nutritional information based on data collected from complementary sensors.

2.2. Approaches for Nutritional Information Capture

Given the increase in food-related health problems [22], there are several research to develop new techniques to enable users to measure and analyze their food intake, which we review them briefly in the following.

2.2.1. Food Intake Detection. Approaches in this research area focus on detecting whether user activities could be classified as eating, without attempting to recognize what types of foods are consumed. Based on the observation of user daily activities, systems in this category are able to identify users' habits and can provide some advice about their food habits in order to assist in healthcare [23].

As a method that applies biting detection, Scisco et al. [24] introduces an approach where users wear a sensor (gyroscope) on their wrists with the purpose of detecting biting movement as a way of detecting food consumption.

Farooq and Sazonov [25] proposed an approach for detection of chewing movement. An accelerometer, a hand to mouth sensor, and a piezoelectric sensor are integrated into a wearable device, which is used to collect signals from the user's movements. Data analysis is based on pattern recognition using different assemble classifiers. Kalantarian et al. [26] employed a piezoelectric sensor equipped in a wearable necklace to detect the movement of the throat and determine whether users are eating or not. The device is able not only to detect food intake but also to provide an estimation on the volume of food consumed and to classify simple types of food.

Cheng et al. [27] focus on detecting certain actions based on the effect of capacitance from electrodes installed inside user's clothes. The approach is able to detect different activities, including not only swallowing and chewing foods but also talking or head movements.

Limitations of these approaches are twofold. First, they can only indicate whether users are eating or not; these systems cannot detect what is being consumed by users. Second, these approaches require users to wear devices that can be invasive or uncomfortable, and thus can limit their interest in adopting the approach. Our proposed approach, on the other hand, has non-invasiveness as one of its key design principles, what we expect may encourage users adoption and the success of the application.

2.2.2. Nutrition Monitoring Systems. Research in this category aims at developing systems that can detect what is being consumed by users, rather than just determining whether users are eating or not. Thus, these systems have more potential to generate information that can be relevant for dietary intake management. Approaches related to our work focus on automatically monitoring and detecting user activities by using IoT sensors.

In recent years, there have been a variety of proposed approaches in developing nutrition monitoring system with the help of sensor devices that are attached to the human body to obtain more reliable information from users and improve the accuracy in dietary assessment [23]. In such *Wearable approaches*, there are many sensing prototypes proposed which are applied in different position of the human body such as ears [28] and teeth [29]. Sun et al. [30] proposed a wearable device called eButton that looks like a decorative button but contains various sensors, including two cameras, a UV sensor, a proximity sensor for motion observation of hand or arm, and a GPS for detecting the current geographical location of the user. This wearable device is capable of detecting if the user is having a meal and can capture and store images of the food. However, this system raises privacy issues due to the problem of monitoring and pictures capture and storage without users' awareness.

Environmental approaches are based on sensors that are not attached to the human body, hence reducing the intrusiveness of the system. However, the accuracy of these systems is usually not high and they are difficult to be used in the practice [23]. Gu and Wang [31] and Zhou et al. [32] developed solutions that are examples of such environmental sensors for monitoring nutrition of users based on consumed foods. Zhou et al. [32] proposed a system for dietary assessment using a smart table cloth that measures the volume of foods and recognizes the type of food based on eating behavior of users with approximately 80% of accuracy.

In *Standalone approaches*, sensors used in nutrition analysis system are neither attached to the human body nor rely on the surrounding environment. In this approach, some external devices will be integrated with smartphones to enable users to directly do the measurement of food items and receive the nutritional information [33]. For instance, users can use a handheld NIRONE spectrometer sensor from Spectral Engines[1] to receive a fast response about nutritional values of food items. This approach only adopts limited number of sensors which mainly suitable for homogeneous foods.

Limitations of these approaches include high intrusiveness of wearable approaches and low accuracy of environmental approaches, which motivated us to look for more holistic solutions that apply standalone approaches and heterogeneous food recognition to increase accuracy without being intrusive to users.

2.2.3. Food Recognition. This section discusses research in food recognition that can be applied in the nutrition monitoring area. According to the definition of *food computing* from Min et al. [34], this area of research can be separated into three different types of approach in detecting and recognizing foods, including single label approaches, multiple labels approaches, and sensors approaches.

Single label approaches focus on the simplification in the real-world scenario, which considers only one food per image input. Using this approach, Yang et al. [35] proposed a new approach for food recognition by measuring the values of pairwise visual features. These values are then represented in a histogram and converted to vectors with multiple dimensions for the purpose of classification. By applying appropriate classifiers on these vectors, nearly 80% of accuracy was achieved, which was better than existing methods.

Multiple labels approaches address the issue that usually users take only one picture of their meals that includes all foods as the input for the nutrition application, and thus single label approaches, which expect only one food in the image, cannot handle these cases. In the area of multiple label approaches, Matsuda et al. [36] proposed a method for classifying foods in an image with multiple food items by detecting the region of each candidate. After testing on ten food items, the method achieved approximately 55.8% of accuracy. Kagaya et al. [37] proposed a method using convolutional neural networks (CNN) for recognizing foods. Experiments presented did not allow a direct comparison between approaches.

Sensors and mobile based approaches leverage the fast development of mobile computing and IoT in recent years, which enabled food recognition not to rely solely on computer vision anymore. Besides computer vision approaches using mobile applications with deep learning techniques [38], [39], IoT sensors also join in this field as a different source for food recognition approaches. For example, based on the idea of acoustic recognition, Gao et al. [40] proposed an application named iHearFood with the purpose of detecting the activity of chewing foods based on sounds captured by Bluetooth headsets. Depending on the categories of sounds, the system can also recognize what type of food users are consuming. In order to accomplish that, authors also proposed a technique for food classification based on

1. https://www.spectralengines.com/

deep learning, which results in accuracy between 77% and 94%. Mirtchouk et al. [41] proposed a method that combines both acoustic-based and motion-based approaches to detect and recognize foods. The proposed method uses as input chewing sounds and movements from the wrist to improve the accuracy of food classification. As a result, by testing on 40 different food items, the method achieved accuracy of around 82.7%.

As discussed in the introduction, approaches based on mobile computing and IoT have the disadvantage of being constrained by the resources available in the mobile devices. Although smartphones are powerful enough to handle the required computation of food recognition, this occurs at a cost in terms of battery life, what can hinder adoption of the approach. Edge computing helps in alleviating this issue, and therefore it is explored in this paper. Our previous approach [11] employed a number of IoT sensors and cloud processing for monitoring nutrition of users, but without utilizing edge to reduce latency and costs related to cloud usage. Based on the findings of this early architecture, the system discussed in the next sections leverages edge to increase the capabilities of the system.

3. The Proposed System

Healthy eating habits are considered a major factor for people to maintain a healthy life. Unawareness of what constitutes healthy eating can cause health problems, obesity or overweight. According to Nordström et al. [42], the number of people becoming obese is increasing every day due to their unhealthy eating habits, and these habits are considered one of the main factors for various diseases such as cardiopathy and diabetes, which results in major costs in the healthcare system.

Because of that, there is a need for nutrition monitoring systems that maintain and improve people's health by monitoring what types of foods they are eating daily or in a fixed period. In order to improve public health, these systems need not only to monitor and analyze different types of foods, but also provide necessary advice to users about their meals. With these considerations, we developed an initial nutrition monitoring system [11]. The start point of this approach is a *Smart Scanner* where users deposit their dish with the food. The Smart Scanner contains a smart scale, and an array of cameras. It recorded the weight of the dish and uploaded the photos and a timestamp to a cloud server. The photos enabled a 3D model of the dish to be asynchronously generated on a High Performance Computing (HPC) system in the backend, and this model was also stored in the cloud server.

Nevertheless, we identified a number of limitations in the early design, namely: (i) the 3D model requires expensive software and hardware to compute, and delivered little value in terms of the information provided. (ii) it required manual identification of the food, as there was no step to detect the food being consumed; and (iii) at the sensor layer of the solution, it was heavily dependent on the particular proprietary method for interaction with sensors. New sensors required new code to adapt the sensor to the Smart Scanner, rather than relying in standard communication models.

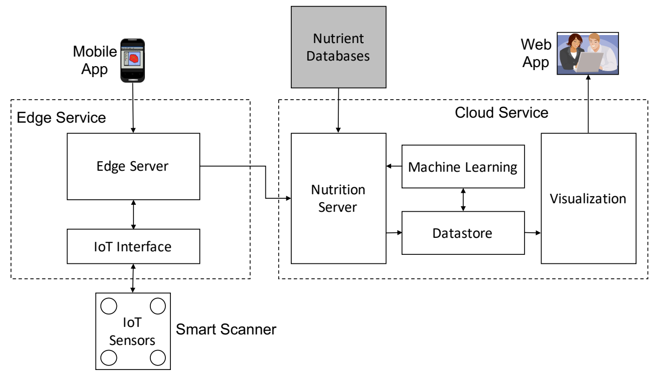

Figure 1. System architecture.

With the above observations at start point, we realized that a more generic solution required also a better abstraction in the sensor layer, removal of HPC resources of the solution, and methods for capture of more information. These considerations guided the requirements and design of the proposed system, as detailed in the rest of this section.

3.1. Requirements and Design Considerations

Our Smart Nutrition Monitoring System has been designed with the following requirements in mind:

- The system must collect nutritional information from foods and store in a persistent database. To enable a detailed analysis from user data over time. This data must be suitable to be consumed by end users directly, but could also aggregated in reports that are relevant for dietitians and other health practitioners,
- The system must not require any wearable sensor, either in the clothes or in the user's body.
- Other than authentication information, no other direct input from the user should be required.
- Minimization of the use of cloud backend services to reduce ongoing costs when operating the solution.
- Standardized interfaces should be used as much as possible to facilitate extensibility of the solution.

Some of the points above were supported by our original design [11], and therefore were kept. This includes the use of a Smart Scanner for collection of food data (although a different design and configuration of sensors has been used at this time), use of RESTful APIs supporting the backend, the use of Raspberry Pi boards to aggregate the data of the sensors, and the use of a mobile phone to provide authentication to the platform. However, a re-design was required to enable the use of standards at the sensor layer, to enable automatic food identification and to leverage local computing capability as much as possible while also connecting to a cloud backend to provide complementary processing capacity, thus transforming the solution in a edge computing application providing nutrition monitoring service. Next, we detail the architecture of such system.

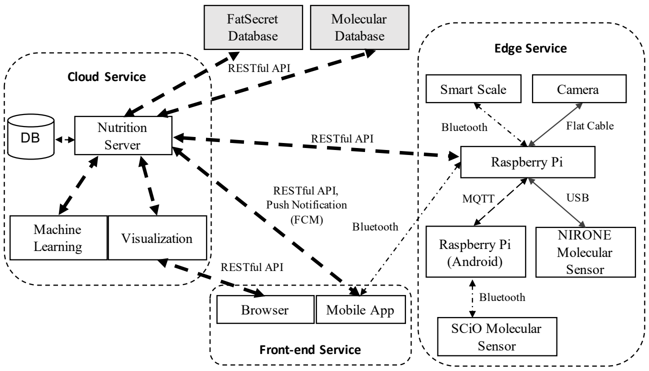

Figure 2. System implementation.

3.2. System Architecture

Figure 1 presents the proposed loosely coupled architecture based on mobile edge computing.

The *edge service* is the central component of the architecture. It interfaces with mobile devices, smart scanner, and the cloud services. Internally, this service contains two components. The first is the IoT Interface module, whose role is to interface with sensors in a number of existing interfaces (both wireless such as Bluetooth and MQTT, and wired such as USB). The IoT devices interact with the second component, the edge server. The edge server is responsible for authorizing users to scan, aggregate input from sensors and submit for the cloud for storage, and trigger the food recognition and analysis.

The *cloud service* supports two types of activities. The first are activities that are computationally demanding to be executed in mobile devices or in the edge, but at the same time are not latency-sensitive, such as the machine learning for food detection. The second type of activities is related to data storage and visualization. The nutrition server is responsible to communicate to the edge service and store the data in the database. Machine learning component is a model for food recognition where it uses the food image as an input and provides the list of ingredients as the output to the nutrition server. These ingredients will be translated to the nutritional information using API calls to external nutrient databases.

The system architecture has interface to the end users through different services. The main interface is a mobile application which enabling registration and access to the smart scanner. The user authentication will be handled using edge service's API through the mobile app. The other component of the system interface is a web application which provides data visualisation to users.

3.3. System Implementation

Figure 2 presents the system implementation based on the proposed architecture. It is broadly organized around three services: front-end service, cloud service, and edge service. These services interact which each other via RESTful APIs, which are explained in the following.

Frontend service: our frontend service has been implemented in the form of a mobile app. It has been implemented as an Android native app for the purposes of validation. As it needs to communicate with the edge service via Bluetooth connection, the app requires access to the device's Bluetooth, and other secondary permissions associated with that (such as coarse location information). Bluetooth was preferred over WiFi to enforce physical proximity between users and edge devices, as discussed later. In addition, a web application using HTML 5 and CSS has also been developed to support data visualization for users using standard browsers.

Cloud service: it implements four core components, namely Nutrition server, Machine learning, Database and Visualization component. The Nutrition server stores and processes data received from the edge service, thus enabling tracking of user nutritional information and generation of reports, which can be used by the Visualization component. It is also responsible for transferring data to the machine learning component for food recognition. Finally, this service is also responsible for storing user data. Besides offering RESTful APIs for interaction, the Nutrition server can also push notifications to the frontend service's mobile App to provide nutritional information to the user. This service has been implemented using Spring MVC[2] and Java. The database is MySQL. Information about users that stored in the database at registration time (can be updated) are name, email, date of birth, gender, weight, and height. The visualisation service is implemented based on Node.js platform. The user interface is developed using React and Chart.js both available JavaScript libraries to create a graphical interface.

The machine learning component has been implemented using TensorFlow[3]. The best and largest publicly available dataset that we can utilize to establish and develop accurate ingredients recognition model is the Recipe1M dataset from Salvador et al. [43] which contains one million structured cooking recipes and their images. We developed a workflow including several Phyton scripts to generate a new dataset with food images and list of ingredients from their recipes. We trained the model using different random subsets of 100, 1000, 10000 images/recipes pairs from the new dataset. Since the dataset ingredients mean was calculated to be 9 and also reported in Salvador et al. [43], so when selecting a training subset of images with fewer ingredients, the model can detect the ingredient with high accuracy. We then noticed that, as we increased the number of ingredients the accuracy decreases. To investigate this further and in details, we first filtered the recipes from any numbers, characters and unnecessary content. This slightly improved the accuracy, but still requires further analysis and more advanced methods and algorithms to be used or developed for better recognition of the image details, especially for dishes with more or close to the mean number of ingredients.

Edge service: this service is implemented with a Rasp-

2. https://spring.io/
3. https://www.tensorflow.org/

berry Pi serving as the computing power resource for the service. Sensors available in the current implementation include a camera that receives the signal from the edge server to take pictures of foods from users, a smart scale device that measures the weight of foods after receiving the signal from the edge server, and two molecular sensors [33] (i.e., SCiO Sensor and NIRONE Sensor) which are used for detecting specific type of food material such as raw meat or cooked meat. The purpose of the molecular sensor is to provide further detail and improve the accuracy of food recognition and nutrition analysis process, and it requires access to an external API to complement its operation. Regarding communication protocols, our system currently supports MQTT and Bluetooth, which are widely supported across sensors. Because the sensors chosen to be added to the system may not support MQTT or Bluetooth, the system also supports extra Raspberry Pi devices that can be used as a bridge between the chosen sensor and the edge server. When this option is used, the sensor communicates with the bridging Raspberry Pi via Bluetooth (or sometimes the sensor is directly attached to the edge server via one of its interfaces) and the bridging and the edge server communicate via MQTT.

One way to improve the system security was requiring users to be close to the edge device to be allowed to trigger the food scanning. Therefore, we rejected a RESTful API access via wireless networks in favor of Bluetooth-based access for user authentication and triggering scanning process.

Food Scanning process: The whole process of food scanning and data acquisition occurs as follows. Initially, the user deposits the dish containing the food to be measured in the Smart Scanner. Then, using the provided mobile app, the user authenticates with the service using Bluetooth (to restrict the distance between the client and the Smart Scanner). When the user is authenticated, it can trigger the scan process with the app. When the process is triggered, information from the sensors are collected by the edge server and user will be notified to remove the dish. The collected data will be sent to the cloud service for long term storage and analysis. At the same time, a photo taken at the Smart Scanner is uploaded by the edge server to the cloud for food recognition purposes. Once the image is recognized, the cloud service queries the nutrition API (FatSecret[4]) to obtain nutritional information from the food. We also use a molecular database[5] to extract nutritional information from the molecular sensors. The final results will be sent back to the user mobile app via push notification. Finally, data from the sensors, the photo, and the nutritional information will be stored in the database. Users (or dietitians) can access the data via a web application, which can generate relevant visualizations of the collected data.

4. https://www.fatsecret.com
5. https://www.consumerphysics.com/

TABLE 1. CONFIGURATION SETUP.

Item	Module	Specifications
Mobile	Android Smartphone	1.9Ghz octa-core Exynos CPU, 2GB RAM
Edge	Raspberry Pi Model B	1.4Ghz quad-core ARM CPU, 1GB RAM
Cloud	AWS EC2 Instance	t2.medium, 2 vCPUs, 4GB RAM
ML	AWS EC2 Instance	p2.xlarge, 4 vCPUs, 1GPU, 61GB RAM
Sensor 1	Camera	Raspberry Pi 8MP Camera
Sensor 2	Scale	SITU Smart Scale
Sensor 3	SCiO Sensor	Molecular Sensor 700-1100nm
Sensor 4	NIRONE Sensor	Molecular Sensor 1750-2150nm

4. Performance Evaluation

In this section, we present the results of performance evaluation to measure service time and power consumption in the implemented system. The configuration setup and specifications of modules in the experiments are listed in Table 1. We used Amazon Web Services (AWS) for cloud services including an EC2 instance for the nutrition server and a GPU instance for machine learning (ML). The communication network between mobile and the edge is a Bluetooth connection and 30Mbps WiFi connection between edge and cloud services.

We have conducted two sets of experiments. In the first set, we measured the service time in each components of the implemented system to show how it can be responsive to users. In the second set, we measured the power consumption on the mobile phone using Trepn Profiler Tools[6] as the primary interface to the user. Each experiment has been conducted 30 times and average results are reported.

Results of the service time experiments are reported in Table 2 and Table 3 for edge and cloud components, respectively. As can be seen in Table 2, scanner time is the total time for the edge component to collect the information from various sensors. Since we used multi-threading model in the edge, the scanner time is much smaller than the aggregated data collection time of all sensors. Thus, the total time that a food item needs to be kept in the Smart Scanner is less than 10 seconds. We also measured the total turnaround time on the mobile app which is 10.81 seconds, which is a reasonable time for users to scan the food before start eating. It should be noted that sending data to cloud is not included in the scan time as this occurs asynchronously in the background.

As soon as the cloud component receives the scan data, it uses the machine learning model for food detection and then external APIs to extract the nutrition information of each particular ingredient item. For these experiments, we trained the model with 1000 sample images and then used 100 testing images for evaluation of the model. We observed accuracy of 83% for single ingredient detection. The inference time was about 2.15 seconds as listed in Table 3. The time for model training is not relevant in this experiments because it occurs in the background as a batch operation, and the training is completely independent from the inference (i.e., once the training completes, it can be used for an

6. https://trepn-profiler.en.uptodown.com/android

TABLE 2. EDGE SERVICE TIMING (SECONDS).

Scanner	Camera	Scale	SCiO Sensor	Upload to Cloud
9.85	3.35	6.92	4.79	11.26

TABLE 3. CLOUD SERVICE TIMING (SECONDS).

Machine Learning	SCiO Analysis	FatSecret API	DB update
2.15	3.46	0.45	3.35

TABLE 4. MOBILE POWER CONSUMPTION (WATT).

Mobile Edge	Mobile Cloud
2.80	8.11

arbitrary amount of time until it is updated). Depending on the type of the food item, we use SCiO Cloud API or FatSecret API to get the nutrition information (i.e, fat, protein, hydrocarbon) and inserting them into the database. The total time for the cloud service is 9.41 seconds which is how long it takes for a user to receive a notification including the nutritional information on the mobile app.

In the second experiment, we measured the power consumption of the mobile phone in two different application configurations including mobile edge and mobile cloud. In the mobile edge configuration, as depicted in Figure 2, we have a lightweight mobile app which has a user interface to users and capability to communicate to the edge service. In the second configuration, we removed the edge service and ran the whole service on the same mobile phone as a mobile cloud configuration. Results of these measurements are reported in Table 4 where it can be seen that the mobile cloud configuration consumes 3 times more power than mobile edge. This reveals that, while mobile phones are able to run various applications, power consumption remains as one of their limitations. So, mobile edge computing can address this issue by offloading the computations and communications to the edge. Moreover, this could help users to use less storage on their mobile phones and improve resource efficiency in long term.

The last but not least, edge services will enable users to have more flexibility and scalability in terms of applications and services. For the smart food scanner, we adopted several sensors which interfacing them with a mobile phone is challenging, or even impossible for different versions of phones. Mobile edge computing enables us to improve and modify the food scanner system with no impact on user devices and their capabilities.

5. Conclusions and Future Work

In this paper, we designed and implemented a novel smart nutrition monitoring system using a food scanner. We proposed a system architecture based on heterogeneous IoT sensors and mobile edge computing for food data collection and analysis on the cloud. We implemented the system using several sensors to demonstrate that this technology is practical, non-invasive, and has minimum participants' burden. Moreover, the developed system consumes much less mobile resources in terms of computation, communication and storage so could be extended independent of mobile phone capabilities. We aim to continue working on this system by developing new machine learning models such as modified or special segmentation deep learning models and using Region of Interest for training the system for better accuracy to detect dishes with different number of ingredients. Implementing the cloud services using Serverless functions [44] will be another future work. We also aim to add a voice-controlled interface to the system so users are able to communicate to the system even without using the mobile phones.

Acknowledgments

This work was partially supported by the AWS Cloud Credits for Research program.

References

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, Jun 2009.

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, Sep 2013.

[3] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, Apr 2014.

[4] X. Chen and X. Lin, "Big data deep learning: Challenges and perspectives," *IEEE Access*, vol. 2, pp. 514–525, 2014.

[5] Akamai, "Akamai online retail performance report: Milliseconds are critical," https://www.akamai.com/uk/en/about/news/press/2017-press/akamai-releases-spring-2017-state-of-online-retail-performance-report.jsp. Accessed on 15-Aug 2019.

[6] B. Zhang, N. Mor, J. Kolb, D. S. Chan, K. Lutz, E. Allman, J. Wawrzynek, E. Lee, and J. Kubiatowicz, "The cloud is not enough: Saving IoT from the cloud," in *7th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 15)*. USENIX Association, 2015.

[7] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan 2017.

[8] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 25 Dec 2013.

[9] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, Feb 2018.

[10] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for the internet of things with edge computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, Jan 2018.

[11] B. Javadi, R. N. Calheiros, K. M. Matawie, A. Ginige, and A. Cook, "Smart nutrition monitoring system using heterogeneous internet of things platform," in *Internet and Distributed Computing Systems*, G. Fortino, A. S. Ali, M. Pathan, A. Guerrieri, and G. Di Fatta, Eds. Springer International Publishing, 2018, pp. 63–74.

[12] S. Nastic, H. Truong, and S. Dustdar, "A middleware infrastructure for utility-based provisioning of IoT cloud systems," in *2016 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2016, pp. 28–40.

[13] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost efficient resource management in fog computing supported medical cyber-physical system," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 1, pp. 108–119, Jan 2017.

[14] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song, and M. Qiu, "A scalable and quick-response software defined vehicular network assisted by mobile edge computing," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 94–100, Jul 2017.

[15] Z. Georgiou, M. Symeonides, D. Trihinas, G. Pallis, and M. D. Dikaiakos, "StreamSight: A query-driven framework for streaming analytics in edge computing," in *2018 IEEE/ACM 11th International Conference on Utility and Cloud Computing (UCC)*. IEEE, 2018, pp. 143–152.

[16] C. Chang, S. Narayana Srirama, and R. Buyya, "Indie fog: An efficient fog-computing infrastructure for the internet of things," *Computer*, vol. 50, no. 9, pp. 92–98, Sep 2017.

[17] F. Mehdipour, B. Javadi, A. Mahanti, and G. Ramirez-Prado, "Fog computing realization for big data analytics," *Fog and Edge Computing: Principles and Paradigms*, pp. 259–290, 2019.

[18] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[19] X. Sun and N. Ansari, "Edgeiot: Mobile edge computing for the internet of things," *IEEE Communications Magazine*, vol. 54, pp. 22–29, 12 2016.

[20] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications (INFOCOM'16)*. IEEE, April 2016.

[21] C. Liu, Y. Cao, Y. Luo, G. Chen, V. Vokkarane, M. Yunsheng, S. Chen, and P. Hou, "A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure," *IEEE Transactions on Services Computing*, vol. 11, no. 2, pp. 249–261, Mar 2018.

[22] A. Fardet and Y. Boirie, "Associations between food and beverage groups and major diet-related chronic diseases: an exhaustive review of pooled/meta-analyses and systematic reviews," *Nutrition reviews*, vol. 72, no. 12, pp. 741–762, 2014.

[23] T. Vu, F. Lin, N. Alshurafa, and W. Xu, "Wearable food intake monitoring technologies: A comprehensive review," *Computers*, vol. 6, no. 1, Mar 2017.

[24] J. L. Scisco, E. R. Muth, and A. W. Hoover, "Examining the utility of a bite-countbased measure of eating activity in free-living human beings," *Journal of the Academy of Nutrition and Dietetics*, vol. 114, no. 3, pp. 464–469, Mar 2014.

[25] M. Farooq and E. Sazonov, "Detection of chewing from piezoelectric film sensor signals using ensemble classifiers," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 4929–4932.

[26] H. Kalantarian, N. Alshurafa, and M. Sarrafzadeh, "A wearable nutrition monitoring system," in *2014 11th International Conference on Wearable and Implantable Body Sensor Networks*. IEEE, 2014, pp. 75–80.

[27] J. Cheng, O. Amft, and P. Lukowicz, "Active capacitive sensing: Exploring a new wearable sensing modality for activity recognition," in *Pervasive Computing*, P. Floréen, A. Krüger, and M. Spasojevic, Eds. Springer Berlin Heidelberg, 2010, pp. 319–336.

[28] O. Amft, "A wearable earpad sensor for chewing monitoring," in *The 9th Annual IEEE Conference on Sensors (IEEE SENSORS)*. IEEE, 2010, pp. 222–227.

[29] E. Thomaz, I. Essa, and G. D. Abowd, "A practical approach for recognizing eating moments with wrist-mounted inertial sensing," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'15)*. ACM, 2015, pp. 1029–1040.

[30] M. Sun, L. E. Burke, Z.-H. Mao, Y. Chen, H.-C. Chen, Y. Bai, Y. Li, C. Li, and W. Jia, "eButton: A wearable computer for health monitoring and personal assistance," in *Proceedings of the 51st Annual Design Automation Conference (DAC'14)*. ACM, 2014, pp. 16:1–16:6.

[31] H. Gu and D. Wang, "A content-aware fridge based on RFID in smart home for home-healthcare," in *11th International Conference on Advanced Communication Technology*, vol. 02. IEEE, 2009, pp. 987–990.

[32] B. Zhou, J. Cheng, M. Sundholm, A. Reiss, W. Huang, O. Amft, and P. Lukowicz, "Smart table surface: A novel approach to pervasive dining monitoring," in *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2015, pp. 155–162.

[33] G. Rateni, P. Dario, and F. Cavallo, "Smartphone-based food diagnostic technologies: A review," *Sensors*, vol. 17, no. 6, Jun 2017.

[34] W. Min, S. Jiang, L. Liu, Y. Rui, and R. Jain, "A survey on food computing," *CoRR*, vol. abs/1808.07202, 2018. [Online]. Available: http://arxiv.org/abs/1808.07202

[35] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, "Food recognition using statistics of pairwise local features," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2249–2256.

[36] Y. Matsuda, H. Hoashi, and K. Yanai, "Recognition of multiple-food images by detecting candidate regions," in *2012 IEEE International Conference on Multimedia and Expo*. IEEE, 2012, pp. 25–30.

[37] H. Kagaya, K. Aizawa, and M. Ogawa, "Food detection and recognition using convolutional neural network," in *Proceedings of the 22nd ACM International Conference on Multimedia*. ACM, 2014, pp. 1085–1088.

[38] K. Yanai, R. Tanno, and K. Okamoto, "Efficient mobile implementation of a CNN-based object recognition system," in *Proceedings of the 24th ACM International Conference on Multimedia*. ACM, 2016, pp. 362–366.

[39] S. V. B. Peddi, P. Kuhad, A. Yassine, P. Pouladzadeh, S. Shirmohammadi, and A. A. N. Shirehjini, "An intelligent cloud-based data processing broker for mobile e-health multimedia applications," *Future Generation Computer Systems*, vol. 66, pp. 71–86, Jan 2017.

[40] Y. Gao, N. Zhang, H. Wang, X. Ding, X. Ye, G. Chen, and Y. Cao, "iHear food: Eating detection using commodity bluetooth headsets," in *2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*. IEEE, 2016, pp. 163–172.

[41] M. Mirtchouk, C. Merck, and S. Kleinberg, "Automated estimation of food type and amount consumed from body-worn audio and motion sensors," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 451–462.

[42] K. Nordström, C. Coff, H. Jönsson, L. Nordenfelt, and U. Görman, "Food and health: individual, cultural, or scientific matters?" *Genes & Nutrition*, vol. 8, pp. 357–363, 2013.

[43] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Torralba, "Learning cross-modal embeddings for cooking recipes and food images," *IEEE, Proc. Comput. Vis. Pattern Recogn*, pp. 3020–3028, 2017.

[44] I. Baldini, P. Castro, K. Chang, P. Cheng, S. Fink, V. Ishakian, N. Mitchell, V. Muthusamy, R. Rabbah, A. Slominski *et al.*, "Serverless computing: Current trends and open problems," in *Research Advances in Cloud Computing*. Springer, 2017, pp. 1–20.