IoT Devices Recognition through Object Detection and Classification Techniques

Argyro Mavrogiorgou, Athanasios Kiourtis, Dimosthenis Kyriazis
Department of Digital Systems
University of Piraeus
Piraeus, Greece
{margy, kiourtis, dimos}@unipi.gr

***Abstract*—Internet of Things (IoT) has dramatically penetrated into today's daily routine, with the number of new devices being growing exponentially, anticipating by year 2035 over one trillion of IoT devices to be connected around the world. However, IoT is still in its infant stage, regarding mainly the heterogeneity of the connected devices. In order to contribute to this gap, this paper proposes a mechanism for recognizing heterogeneous IoT devices of unknown device type, by applying specific computer vision techniques. This mechanism implements four (4) stages to successfully recognize a device of unknown nature. Initially the discovery and the connection of both known and unknown devices occur, followed by the online search of their images, so as to categorize and map the unknown devices' images with the images of the known devices. As a result, since the device type of the known devices is a priory known, and each unknown device is mapped with one or more images of these known devices, the device type of each unknown device becomes known, as well. The prototype associated with this paper provides an example of this mechanism, demonstrating in detail each discrete stage.**

Keywords—IoT, heterogeneous devices, unknown devices, device recognition, object detection, object classification

I. INTRODUCTION

Internet of Things (IoT) [1] is a combination of diverse smart objects that contain sensing capabilities, thus offering the ability to interconnect real world objects as smart objects, being able to sense each other. The concept of IoT started back in 1999, when it was envisioned as an ecosystem of hardware that could be able to communicate with other devices. Thus, people could be able to control all these devices so as to improve their society and personal lives [2]. Currently, this ecosystem still exists, addressing the traceability, visibility, and controllability of these devices in various Information Technology (IT) applications, such as the healthcare services, the environmental monitoring, the smart homes, the smart cities, and the security systems [3]. Therefore, IoT supports numerous vertical markets, all of them being connected to the Internet, but not necessarily to each other. The concept of IoT really began to take off in 2015, when a combination of data analytics, high-speed and almost ubiquitous connectivity, and a variety of applications became available on a mass scale [4]. IoT still continues to expand into homes, businesses, and other environments, as more and more devices are purchased and connected to the IoT world, aiming to collect, manage, and share data [5]. More specifically, in 2020 it is anticipated that there will be 50 billion devices worldwide, about 6 per person [6], whereas ARM predicts that by the year 2035 the world will be connected with more than one trillion IoT devices [7]. These numbers are continuously growing, where almost 31 billion IoT devices are estimated to be shipped worldwide, purchased and used until the year 2020 [8].

However, despite its wide spread emergence, IoT is still in its infant stage, demanding a big scope for research in a variety of issues like interoperability, security, heterogeneity of devices, production of tremendous amount of data, integration with existing IT systems and platforms, and so on [9]. Taking into consideration that the major part that constitutes the IoT world is the devices, the main challenges that have to be solved are directly related with such devices. To this context, one of the most challenging problems that has to be anticipated is that all these devices are characterized by a high degree of heterogeneity, thus needing applications to deal with each specific different device, identify and understand its nature, in order to use its properties and data. As a result, all the existing IoT platforms that need this data should contain mechanisms that will be independent of the devices' hardware and software capabilities, being able to dynamically and rapidly identify heterogeneous IoT devices in order to finally gather their data [10]. However, existing IoT platforms lack sufficient flexibility to adapt to these changes, as their recognition and integration techniques are both static and sensitive to new or changing devices' environments [11]. Thus, without being able to identify the different incoming IoT devices, they are not able to decide whether they are related with each specific platform, and their data needs to be collected. Therefore, both devices' recognition and devices' integration are challenging research topics in the IoT era.

Having undeniably recognized the emergence of this problem, many studies have tried to solve it by addressing it through different methods and techniques, such as the ones in [12], [13], and [14]. However, none of these studies has considered the continuously growing field of computer vision, which can be easily adopted to cope with this challenge. To be more specific, computer vision applications seek to extract information about the world and its surroundings, by examining just single digital images or sequences of them. This concept could be easily adopted into the era of the IoT devices, taking advantage of the images that could be captured by the heterogeneous devices in order to be recognized. It is a common phrase that "an image is worth a thousand words" [15]. Indeed, by simply looking at an object, a lot of information can be extracted from it, providing useful insights. Classifying images using human eyes may be slow, requiring a lot of manpower. It was then the idea of feature detection and image classification that came up for recognizing and classifying all these images. Considering all the above, this idea can be effectively applied into the IoT domain for resolving the problem of the IoT devices' heterogeneity that has attracted a lot of attention nowadays.

Henceforth, this paper proposes a mechanism for dynamically recognizing the nature (i.e. device type) of heterogeneous IoT devices of both known and unknown nature by applying computer vision techniques. Through this mechanism it becomes feasible via the information generated by each different incoming unknown device to discover its specific device type, which was unknown in advance. Thus, the mechanism outputs information about whether each device is related with each specific platform that wants to be connected, and as a result whether each device's data is relevant to this specific platform and needs to be collected. In order to achieve that, initially the discovery and the connection of all the available devices occur, so as to retrieve some of the specifications that they contain. In sequel, the categorization of these devices into either the known (in terms of a priory known device type) or the unknown (in terms of unknown device type) category occurs, based upon their existence into an already created devices' registry that contains the specifications of a number of devices of known device type. By the time that this categorization gets complete, the online search of each unknown device's image takes place, based upon its captured specifications. Finally, the classification of each unknown device's image occurs in combination with the images of the known devices that exist in the already created devices' registry. As a result of this classification, the mechanism categorizes and maps each unknown device's image with one or more images of the known devices. Consequently, since the device type of all the known devices is a priory known, and the unknown device's image is categorized and mapped with one or more of these known devices' images, the device type of the unknown device becomes known, as well. All the aforementioned steps are tested and evaluated by implementing the proposed mechanism on a specific use case.

The rest of this paper is organized as follows. Section II describes the study of the state of the art with regards to the fields of object detection and image classification that are mainly used in the proposed mechanism. Section III describes the developed mechanism for connecting and recognizing different heterogeneous IoT devices of unknown device type, Section IV analyses and discusses a use case based upon the proposed mechanism, whereas Section V addresses our conclusions and plans.

II. Related Work

As mentioned before, the aim of this paper is to dynamically connect and recognize heterogeneous IoT devices of both known and unknown nature in the IoT domain. Even though there exist various methods for categorizing and recognizing different IoT devices [16], the current research investigates and takes advantage of the various computer vision techniques that are widely used nowadays for recognizing and classifying different kinds of images. To be more precise, both IoT object detection and image classification are gaining much interest recently and are going to be thoroughly studied in this paper.

A. Object Detection

Object detection [17] is a computer technology connected with computer vision and image processing, aiming to detect objects or instances of a certain class (e.g. fruits, flowers) both in digital images and videos. Therefore, the scope of object detection analysis is to determine the number, location, size, and position of the objects/instances that are located into these images and videos. Its objective is two-phased, including firstly the identification of the objects that are present in the image by identifying their location, and secondly the filtering of the object of attention. There exist various researches that have investigated the application of object detection techniques, indicating that object detection is the basic concept for recognizing objects [18]. Some examples include helping self-driving cars to safely navigate through traffic, spotting violent behavior in a crowded place, assisting sports teams to analyze and build scouting reports, ensuring proper quality control of parts in manufacturing, or calculating the number of people and vehicles in images [19].

Typically, three (3) discrete steps need to be followed for applying object detection techniques [20]: (i) a model needs to be implemented for generating the regions of interest or region proposals in the form of a set of bounding boxes, spanning the full image, (ii) visual features need to be extracted for each one of the constructed bounding boxes and be evaluated, thus determining whether and which objects are present in the region proposals based on the visual features, and (iii) overlapping bounding boxes need to be integrated into a single bounding box. These steps are applied in multiple approaches that can be used for detecting objects in images, including the commonly used technique of dividing and conquer, which divides a specific image into four (4) equal parts, in order to detect specific objects. The approach of increasing the number of divisions is similar to the previous one, but in that case, the number of patches that are provided as an input to the system is increased. Moreover, there exists the approach of performing structured divisions, where the image is divided into smaller grids and patches of different heights and aspect ratios. Furthermore, there is also the approach of deep learning for feature selection, where the original image is passed through a neural network in order to reduce its dimensions, and a deep learning algorithm is providing predictions as close to the original bounding box as possible [21]. However, among the various object detection approaches that exist, the most common one is the color-based approach, which detects objects based on the color values that they contain. This method is mainly used due to its strong adaptability and robustness. However, it needs improvements on its detection speed, as it requires to test all the possible windows through iterative search and high computational complexity [22].

B. Image Classification

Image classification is often confused with object detection [23]. Generally, image classification is used in the cases that there is need for classifying an image into a certain category, while object detection is used in the cases that there is need for recognizing the location of objects in an image, such as counting the number of instances of an object. Therefore, image classification is responsible for extracting information classes from an image, where the results can be used to create thematic maps. Thus, the objective of image classification is to identify and depict, as a unique either grey or color level, all the contained features of an image [24]. It can be considered as one of the most important aspects in the field of the digital image analysis. Some examples of image classification include organizing personal photos, stocking photography and video, classifying images for websites with large visual databases, recognizing faces on social networks, or advertising based on different targeted audiences [25].

Typically, the steps that are followed during image classification depend on the chosen approach. Henceforth, according to the approach, different training samples and stages can be extracted. To this context, there exist specific methods [26] that can be used to classify images, such as the supervised classification that is used for identifying the examples of the information classes of the various interests that are located into an image. There also exists the maximum likelihood classification in the form of a statistical criterion, which is used for assisting the classification process by finding potential overlapping signatures. Moreover, the minimum distance classification has been developed for classifying images on a database, using a collection of 256 possible class signature segments. In the same notion, there exists the parallelepiped classification, which uses the class limits and stores them in each class signatures in order to determine whether a given pixel belongs to a specific class or not. Finally, the unsupervised classification should be mentioned, which is also a commonly used method for realizing image processing, being able to examine a large number of unknown pixels, and divide them into the corresponding number of classes, based on the natural groupings that are present in an image [27].

C. Object Detection and Image Classificaiton Methods

1) Scale Invariant Feature Transform (SIFT)

SIFT [28] is a commonly used machine vision technique for detecting features in images, by applying four (4) discrete stages. Initially, the scale-space extrema detection is used to detect the different interest points (i.e. keypoints) of an image, whilst the image convolves with the Gaussian filter in its different scales. In sequel, during the keypoint localization, the selection among the different keypoint candidates occurs, by comparing each existing pixel with each other. Sequentially, the orientation assignment takes place, where each pixel is assigned on the image gradient direction. Thus, finally, the creation of the keypoint descriptors occurs, where the locations of the objects that have different orientations and scales are found.

2) Speed up Robust Feature (SURF)

SURF [29] has a similar performance with SIFT, being however much faster than it. More specifically, SURF builds an image pyramid, and filters each layer with the Gaussian filter, by increasing sigma and calculating the difference among the different layers. Like SIFT, SURF looks for the extrema in the difference of the Gaussian filtered versions of the image. This computation is implemented for various image sizes or octaves, and different blurs or scales.

3) Center Surrounded Extrema (CenSurE)

CenSurE [31] is another machine vision technique for detecting features in images, differentiating from SIFT and SURF, as it achieves full spatial resolution at every scale. In more detail, CenSurE uses polygon, hexagon and octagon filters as a more computable alternative in order to circle the filter in an image. Thus, initially, CenSurE computes all the location and scales to find the local extrema in a neighborhood, by simplifying center-surround filter, whilst afterwards, Harris Corner Detector is used for eliminating the entire weak corner.

4) Maximally Stable External Regions (MSER)

MSER [30] is used as a method of blob detection in images. This method is used to find correspondence between two (2) images that have different viewpoints. Hence, MSER is applied to binary images, where all the pixels inside MSER belong to a specific extremal region, taking into consideration the higher or lower intensity with regards to all the pixels on its outer boundary.

5) Haar-like features

Haar-like features [32] implements a classifier that is trained from different images so as to be used in another image for detecting its including objects. Initially, Haar-like features takes as an input images both with faces and without faces, so as to sequentially train the classifier, and then extract the corresponding features. To this end, it should be mentioned that instead of applying all the extracted features at once, it groups them into different stages of the classifier and applies each feature per time.

6) Circular Hough Transformation (CHT)

CHT [33] has been designed to detect circular objects in low-contrast noisy images. It is based on three (3) different parameters, which require larger computation time and memory, thus increasing the complexity to extract the needed information from an image. To reduce this complexity, CHT applications are provided either with a constant value of radius or with a range of radius, prior to running the application.

7) Template matching

Template matching [34] is a machine vision technique for detecting objects from an image that matches a given image pattern. In order to achieve that, the Template matching technique compares the source image with a specific template image or patch, finding the exact matches between them. In the case that this technique concludes that the template image has strong features, it uses the feature-based approach for detecting the contained objects, whilst on the contrary, it uses the template-based approach.

8) Gradient-based method

Gradient-based method [35] is mainly used for estimating the image flow at every position in an image, by implementing spatial and temporal partial derivatives to it. However, there might be some cases that the flow of the image is not known in advance, and as a result it cannot be restricted to a small range of possible values. In order to address this gap, a multi-scale analysis is applied so that the scale of the smoothing prior to the derivatives' estimation can effectively correspond to the scale of the flow.

9) Local Binary Pattern (LBP)

LBP [36] is dividing a window of an image into cells, where each pixel in the cell is compared with its eight (8) adjacent pixels. In the case that the value of the pixel is greater than a neighbor, then the algorithm writes "0", otherwise it writes "1", providing an 8-digit binary number that is converted into decimal. As soon as this process gets complete, the algorithm aggregates the histogram of all the cells by performing calculations and comparisons among them. In sequel, it normalizes the formulated histogram, revealing finally the feature vector of the entire window, which is then used for classifying all the existing images.

D. Object Detection and Image Classification Researches

It is clear that object detection techniques are widely used in several research areas and domains including healthcare, bioinformatics, business, marketing, and agriculture. In this context, multiple researchers have based their work on all of the aforementioned methods, such as the authors in [37] who considered the problem of detecting and localizing generic objects from categories such as people or cars in static images. Hence, they described an object detection system that used mixtures of multiscale deformable part models, in order to represent highly variable object classes. Moreover, based on previous approaches on object detection, the authors in [38], besides repurposing classifiers for performing detection, they framed object detection as a regression problem, in order to gradually separate the different bounding boxes and associate the classes' probabilities. In this context, a single neural network was constructed for predicting all the bounding boxes and classes' probabilities that emerged directly from the corresponding full images. What is more, the authors in [39] presented a cascade-based face detection framework using SURF features. In this framework, the authors dealt with several hundreds of multidimensional local SURF patches instead of hundreds of thousands of single dimensional Haar-like features. As a result, their proposed approach was able to train face detectors within one hour by scanning billions of negative samples on current personal computers. In addition, the authors in [40] proposed a fast-binary descriptor that used visual features, being rotation-invariant and noise-resistant, and thus proving remarkably successful results in several applications, such as object recognition. This descriptor was evaluated through several experiments, being proved as faster and more efficient than SIFT and SURF methods. What is more, the authors in [41] extended SURF so as to be applied in three-dimensional shapes. Moreover, the authors in [42] presented a framework for training a boosting cascade-based object detector from large-scale datasets. Their work concluded that the boosting cascade can be efficiently trained, providing results with high processing speed with regards to object detection. Finally, the authors in [43] proposed a symmetrical SURF descriptor for detecting all the possible symmetrical matching pairs of an image, by applying a mirroring transformation. The most remarkable advantage of their approach was that there was no need for applying background subtraction, as it was extremely efficient for real-time applications.

III. Proposed Approach

Having deeply studied the object detection and image classification techniques, in our approach we proposed a mechanism that could recognize heterogeneous IoT devices of unknown device type, by taking advantage of these existing techniques. To be more precise, the proposed mechanism is dynamically connecting, recognizing, and classifying heterogeneous IoT devices of both known and unknown nature (i.e. device type), based upon their images. This mechanism consists of four (4) stages: (i) Devices Connector, (ii) Devices Identifier, (iii) Devices Image Search Engine, and (iv) Devices Image Classifier, as depicted in Fig. 1. More specifically, in the first stage the discovery and the connection of all the available devices occur so as to retrieve the devices' specifications (i.e. names and MAC addresses). Sequentially, in the second stage, the categorization of these devices into either known (in terms of a priory known device type) or unknown (in terms of unknown device type) type occurs, based upon their existence into an already created devices' registry that contains the specifications of a number of devices of known device type. By the time that this categorization gets complete, the third stage is implemented, where through the Web, the online search of each unknown device's image takes place, based upon its captured specifications. Finally, in the fourth stage, the classification of each unknown device's image occurs in combination with the images of the known devices that exist in the already created devices' registry. As a result, the categorization and mapping of each unknown device's image with one or more images of the known devices occur, identifying the device type of each unknown device.

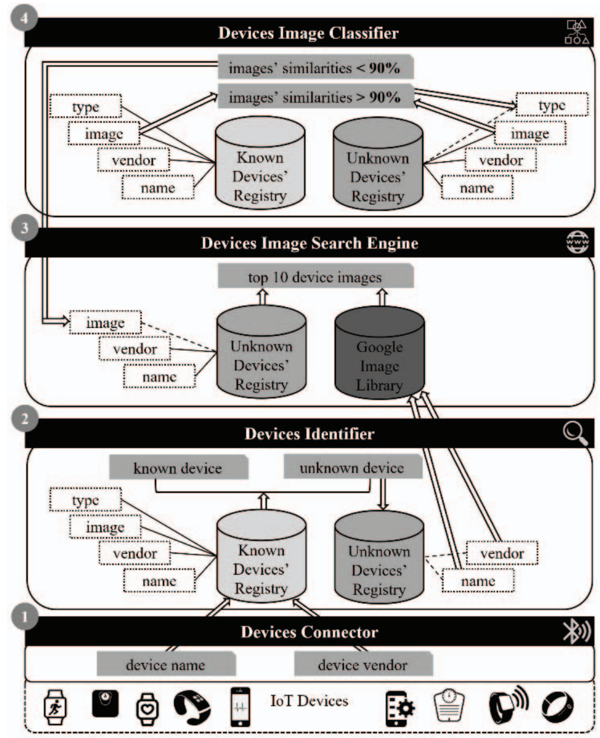

Fig. 1. Overall architecture.

A. Devices Connector

In the first stage of the mechanism, the discovery as well as the connection of the available heterogeneous both known and unknown (in terms of known and unknown device type) IoT devices takes place, followed by the collection of their specifications (i.e. name, MAC address). In more detail, the various available devices (e.g. blood pressure monitors, body weight scales, etc.) are discovered and connected to the mechanism through the implemented Bluetooth interface, which is used for acquiring the devices requested specifications. This interface is based upon the Bluetooth Low-Energy (BLE) [44], which is a core protocol for IoT applications [45]. Therefore, through this interface, the mechanism is able to communicate with all the available Bluetooth-enabled IoT devices that want to be connected to the mechanism.

By the time that all these devices are connected to the mechanism, the latter gathers information about the connected devices' Internet Protocol (IP) addresses. Through the IPs of the devices it becomes feasible to identify each device's specific Media Access Control (MAC) address, which is a unique identifier for it, revealing the Organizationally Unique Identifier (OUI) that allows the association of a MAC address to a specific vendor. This OUI can be easily found through the first three (3) octets of each MAC address, which is then given as an input into the MAC Vendors API [46], through which the mechanism finds the corresponding vendor of each OUI of the corresponding MAC addresses. However, this information is considered poor as it only reveals the name of the vendor, whereas sometimes no organization is associated with the received MAC address. For that reason, apart from the IP address of each device, the mechanism gathers information about the name of each device, so as to use it and combine it with the name of the vendor of the corresponding device in the next steps of the mechanism.

B. Devices Identifier

By the time that all the devices are connected and their specifications (i.e. name and vendor) are captured, the second stage of the mechanism takes place, where the captured specifications are queried and automatically compared with the corresponding specifications of the devices that already exist in a created registry that includes information of known IoT devices (i.e. known devices' registry). Through this way, the mechanism is able to find out whether the connected devices already exist in this registry or not, and thus they will be characterized as either known or unknown accordingly. More specifically, if the specifications of the connected devices are the same with the specifications of the devices that are included into the known devices' registry, then the connected devices are considered to be known, and their device type is automatically recognized. On the contrary, if the comparison concludes that there is no resemblance or even a partial resemblance among the specifications of the connected devices and the specifications of the devices that are included into the known devices' registry, then the connected devices are considered to be unknown, concealing any information regarding their device type. Thus, all the connected devices are characterized to be either of known or of unknown device type.

As soon as this process gets complete, all the devices' gathered information is stored, where the specifications of the unknown devices are stored into a new registry (i.e. unknown devices' registry) that consists of the fields of the device name, and the vendor of each unknown device. Regarding the specifications of the known devices, these do not have to be stored, as they already exist in the known devices' registry, which consists of the fields of the device name, the vendor, the image, as well as the device type of each known device.

C. Devices Image Search Engine

After having successfully stored all the devices' information, the third stage of the mechanism occurs, implementing the online search of the connected unknown devices' images, based upon their captured specifications (i.e. name and vendor). In more detail, the constructed Image Search Engine of the mechanism uses as an input both the name and the vendor of each connected unknown device. Based upon the words that those two (2) specifications contain, it encapsulates them into a query, constructing the corresponding search query in order to use it as an input for its online search via the Web. In our case, since the mechanism gets advantage of the images of the connected devices, the Google Image search engine [47] is used for finding each unknown device's corresponding image. Consequently, the constructed Image Search Engine searches and finds in the Google Images' library each unknown device's image.

As soon as this search gets complete, the Image Search Engine returns the results' list that contains the first top 10 most frequently appeared images. Thus, all the connected unknown devices' images that are found, are stored into each corresponding device's image field that exists in the unknown devices' registry. It should be noted that it is assumed that the constructed Image Search Engine is always returning a list of results, since today's search engines have crawled the Web and found documents to add to their searchable indexes. What is more, they keep the history snippets even if a product is out-of-date without having any correlations with none of the existing webpages. As a result, at least an image of a device will always be available online in the Web.

D. Devices Image Classifier

As soon as the image of each connected unknown device is found and stored into the unknown devices' registry, the fourth stage of the mechanism occurs, implementing the classification of both known and unknown devices' images, so as to finally recognize the device type of the connected unknown devices. In order to achieve that, the proposed mechanism uses the open source software library of Open Source Computer Vision (OpenCV) [48] so as to find the similarities among the different existing images of the devices (both known and unknown). More specifically, OpenCV contains more than 3000 methods for addressing the different existing aspects of computer vision.

The current mechanism takes advantage of the SURF method [49] in order to perform feature detection on the input image (i.e. unknown device's image) and the sample images that exist (i.e. known devices' images), so as to extract the different existing features (i.e. points of interest) of each image in combination with their corresponding descriptors. Thus, by comparing the descriptors obtained from the different sample images and the input image, the different matching pairs are found. It should be mentioned that SURF is used since it is a local feature detector and descriptor that can be effectively used for undertaking tasks such as object detection or classification, which are primarily examined in the proposed mechanism. To be more specific, object detection using SURF is scale and rotation invariant, making it a very powerful technique in comparison with other existing object detection techniques [50]. After the implementation of SURF, a loop is performed through all the sample images to find the best match that they have with the input image, using the Fast Approximate Nearest Neighbour Search (FLANN) feature matcher [51]. In more detail, FLANN is used in the proposed mechanism since it provides a set of algorithms that are optimized for fast nearest neighbor search in large datasets that are of high dimensional features (e.g. images) [52]. Therefore, in our case, since the mechanism needs to cope with the matching and classification of different images, the usage of FLANN is considered to be suitable enough for that purpose. In order to achieve this classification, among FLANN's available algorithms, the mechanism exploits the k-Nearest Neighbors (KNN) matching algorithm [53] in order to extract the best matches of the features of the input image (i.e. unknown device's image) in query with the features of the sample images (i.e. known devices' images).

By the time that this process gets complete, the mechanism outputs the final results, indicating whether there were found any similarities with the image of the connected unknown device (i.e. input image), and the images of the known devices (i.e. sample images). In order to decide whether the calculated percentage of similarity is sufficient for considering the results as reliable, the threshold of 90% is set. In the case that similarities are observed among these devices, and the calculated percentage of similarity is better than the set threshold (i.e. >90%), then the results cite the name and the type of the known device whose image has the greatest degree (i.e. percentage) of similarity with the image of the unknown device. On the other hand, if the mechanism outputs that the greatest calculated percentage of similarity is lower than the set threshold (i.e. <90%), meaning that it did not find sufficient similarities among the unknown and the known devices' images, then a feedback loop takes place back to the third stage of the mechanism (i.e. Devices Image Search Engine), in order to obtain the second image of the results' list.

Since the results' list contains the top 10 most frequently appeared images, all this process is repeated until the unknown device's image has a similarity degree (>90%) with one or more of the known devices' images.

Therefore, as soon as the calculated percentage of similarity is better than the set threshold, since the device type of all the known devices is a priory known, and the unknown device is identified to be identical or almost identical with the corresponding known device, the device type of the unknown device becomes known, as well. Consequently, each connected unknown device is considered as known, providing detailed information about its name, vendor, image, and device type. Thus, each one of these connected recognized devices in combination with their corresponding specifications and images are finally stored in the registry of the known devices' registry, to be considered as known devices for future usage. On top of this, since the final goal of the mechanism is to determine whether each incoming unknown device is related with each specific platform that wants to be connected, and as a result whether its data is relevant to the platform and needs to be collected, based on the results we can conclude whether this data will be kept, depending on each device's recognized device type.

IV. Use Case

A. Use Case Description

In order to perform a complete testing and evaluation of the developed mechanism, 31 IoT medical devices were chosen that could communicate through Bluetooth with the mechanism. In more detail, 30 of them were of known device type, whereas 1 of them was of unknown device type. For the 30 devices we had prior knowledge about their image and specifications (i.e. name and vendor), outlining the type of the device that each one of them belonged to (i.e. activity tracker, blood pressure monitor, body weight scale, glucometer, thermometer, toothbrush, and pulse oximeter). In more detail, 6 of these devices were blood pressure monitors, 2 were glucometers, 6 were body weight scales, 9 were activity trackers, 3 were thermometers, 3 were toothbrushes, and 1 was pulse oximeter. Regarding the 1 device, we had only prior knowledge about its specifications (i.e. name and vendor), since its image and device type were unknown.

TABLE I. Devices' Specifications and types

#	Name	Vendor	Type
1	Fitbit Flex	Fitbit	Activity tracker
2	Withings BPM	Withings	Blood pressure monitor
3	iHealth Ease	iHealth	Blood pressure monitor
4	Oral-B 7000	Braun	Toothbrush
5	ARA toothbrush	Kolibree	Toothbrush
6	iHealth Smart	iHealth	Glucometer
7	iHealth Lite	iHealth	Body weight scale
8	Omron M3 IT	Omron	Blood pressure monitor
9	Fitbit Charge 2	Fitbit	Activity tracker
10	iHealth Wave	iHealth	Activity tracker
11	GentleTemp 521	Omron	Thermometer
12	Oral-B Genius	Braun	Toothbrush
13	Omron BF511	Omron	Body weight scale
14	iHealth Align	iHealth	Glucometer
15	Fitbit Zip™	Fitbit	Activity tracker
16	Withings Thermo	Withings	Thermometer
17	iHealth Air	iHealth	Pulse oximeter
18	Fitbit Aria®	Fitbit	Body weight scale
19	Omron BP710	Omron	Blood pressure monitor
20	iHealth Sense	iHealth	Body weight scale
21	Withings Steel	Withings	Activity tracker
22	Garmin Vivosport	Garmin	Activity tracker
23	iHealth Thermometer	iHealth	Thermometer
24	Withings Pulse O2	Withings	Activity tracker
25	iHealth Track	iHealth	Blood pressure monitor
26	Garmin Vivofit	Garmin	Activity tracker
27	Fitbit Alta HR	Fitbit	Activity tracker
28	Withings Body+	iHealth	Body weight scale
29	iHealth Clear	iHealth	Blood pressure monitor
30	Withings Body Cardio	Withings	Body weight scale
31	TaiDoc TD-3128	TaiDoc	Unknown

B. Experimental Results

The proposed mechanism was implemented in Java SE using the NetBeans IDE v8.0.2 [54]. On top of this, it used a processing environment with 16GB RAM, Intel i7-4790 @ 3.60 GHz x 8 CPU cores, 2TB storage, and Windows 10 operating system. The results of the mechanism are depicted below, following the four (4) discrete stages explained in Section III.

Devices Connector. Initially, in the first stage, 1 device asked for permission to be connected to the mechanism (i.e. the last entry highlighted with grey in Table I), whereas 30 devices already existed in the known devices' registry (i.e. the first 30 entries of Table I). Thus, as soon as the 1 device was connected to the mechanism, its specifications (i.e. name and IP address) were captured. However, as mentioned in Section III, from the device's IP address, we could find the corresponding MAC address, from which we extracted the first three (3) octets, so as to reveal the OUI of the device, and associate the corresponding MAC address to its specific vendor. In order to achieve that, the mechanism used the MAC Vendors API so as to find the corresponding vendor of the OUI of the device's MAC address. As a result, the MAC Vendors API revealed that the connected device's vendor was the TaiDoc, whereas the device's name was TaiDoc TD-3128.

Devices Identifier. Sequentially, in the second stage, both the name and the vendor of the 1 connected device were compared with the corresponding specifications (i.e. name and vendor) of the 30 devices that already existed in the known devices' registry, concluding that the connected device was not included in this registry. Thus, this device considered to be of unknown device type, and its specifications were stored into a new registry, the unknown devices' registry.

Devices Image Search Engine. Afterwards, in the third stage, the online search of the connected unknown device's images occurred, where the constructed Image Search Engine used as an input both the name and the vendor of the connected unknown device. Based upon the words that these two (2) specifications contained (i.e. name: TaiDoc TD-3128 and vendor: TaiDoc), the Image Search Engine encapsulated them into a query, and constructed the corresponding search query in order to use it as an input for the online search through the Google Image library. More specifically, the constructed search query for searching the connected unknown device's image was formatted as "search engine/search?hl=en&q=%TaiDoc+TD-3128&TaiDoc&btnG=Search", where (q) was the start of the query, the mark (?) indicated the end of the URL, (&) separated the arguments among each other, the plus mark (+) represented a space, and btnG=Search denoted that the search button was pressed on the web interface. By the time that this search got complete, the Image Search Engine returned the results' list that contained the first top 10 most frequently appeared images of the connected device, which were stored into the corresponding device's image field of the unknown devices' registry.

Devices Image Classifier. Finally, in the fourth stage, the classification of the 30 known and the 1 connected unknown devices' images occurred, so as to finally identify the unknown device's type. For that purpose, the OpenCV library was used, where initially SURF was implemented for performing feature detection upon the input image of the 1 connected unknown device and the sample images of the 30 existed known devices. Thus, SURF extracted from all these images their different features (i.e. points of interest) in combination with their descriptors, in order to compare the descriptors obtained from each different sample image and the input image, and find the different matching pairs. Therefore, a loop was performed through all the sample images of the 30 existed known devices in order to compare the descriptors that were obtained from their different images and the input image, thus finding the different matching pairs among them. The image of the 1 connected unknown device (i.e. TaiDoc TD-3128) and its corresponding features that were found, are depicted in Fig. 2.

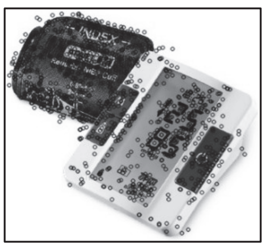

Fig. 2. TaiDoc TD-3128 features.

After the implementation of SURF and the extraction of all the possible matching pairs, so as to find the best match among all the constructed matching pairs, the FLANN algorithm was implemented. More specifically, the KNN matching algorithm was used for finding the best matches of the features of the input image in query with the features of the sample images. As soon as this process got complete, the mechanism output the final results, indicating the percentages of similarity among the 30 existing known devices and the 1 connected unknown device's images. Based on the produced results, the best matching image of the known device that was found along with the image of the 1 connected unknown device was the Omron M3 IT, as depicted in Fig. 3.

Fig. 3. Best matching result of TaiDoc TD-3128.

In more detail, all the captured results are depicted in Table II, indicating the exact number of matching features, and as a result the overall similarity percentage, that each known device's image had with the connected unknown device's image. To be more specific, during the implementation of SURF, it was recognized that the connected unknown device's image contained 332 discrete features, pointing out that a known device's image would have 100% overall similarity with the unknown's one, in the case that it would have 332 features in common with the unknown device.

TABLE II. OVERALL PERCENTAGES OF SIMILARITY

#	Name	Matching Features	Overall Similarity (%)
1	Fitbit Flex	3	0.9
2	Withings BPM	34	10.2
3	iHealth Ease	97	29.2
4	Oral-B 7000	8	2.4
5	ARA toothbrush	10	3.0
6	iHealth Smart	19	5.7
7	iHealth Lite	3	0.9
8	Omron M3 IT	325	97.9
9	Fitbit Charge 2	5	1.5
10	iHealth Wave	6	1.8
11	GentleTemp 521	23	6.9
12	Oral-B Genius	7	2.1
13	Omron BF511	47	14.15
14	iHealth Align	3	0.9
15	Fitbit Zip™	5	1.5
16	Withings Thermo	6	1.8
17	iHealth Air	8	2.4
18	Fitbit Aria®	28	8.4
19	Omron BP710	280	84.3
20	iHealth Sense	6	1.8
21	Withings Steel	6	1.8
22	Garmin Vivosport	4	1.2
23	iHealth Thermometer	5	1.5
24	Withings Pulse O2	11	3.3
25	iHealth Track	301	90.7
26	Garmin Vivofit	5	1.5
27	Fitbit Alta HR	5	1.5
28	Withings Body+	31	9.3
29	iHealth Clear	203	61.1
30	Withings Body Cardio	28	8.4

Based on the results of Table II, it is observed that different percentages of similarities were found among each known device's image and the connected unknown device's image. However, the images of the known devices iHealth Ease, iHealth Clear, Omron BP710, iHealth Track and Omron M3 IT, had the greatest degree (i.e. percentage) of similarities with the unknown device's image. Some of these percentages exceeded the set threshold of 90%, whereas some others did not exceed it. The top 5 best matching results between the connected unknown device (i.e. TaiDoc TD-3128) and the 30 existing known devices are illustrated in Fig. 4.

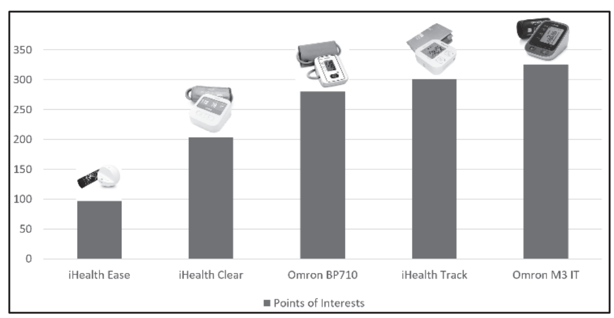

Fig. 4. Top 5 best matching results of TaiDoc TD-3128.

Therefore, since the calculated percentage of similarity between the Omron M3 IT and the TaiDoc TD-3128 was better than the set threshold (i.e. 97.9%), far exceeding it, and the device type of the Omron M3 IT was known that is a blood pressure monitor, the unknown device of TaiDoc TD-3128 was identified to be a blood pressure monitor, as well. Thus, the TaiDoc TD-3128 specifications and image are finally stored in the registry of the known devices' registry, so as to be considered as a known device for future usage. To this end, it should be mentioned that since the calculated percentage of overall similarity between the connected unknown device (i.e. TaiDoc TD-3128) and one of the known devices (i.e. Omron M3 IT) exceeded the set threshold, a feedback loop did not have to take place back to the third stage of the mechanism (i.e. Devices Image Search Engine), in order to obtain the second image of the unknown device's results list, and so on.

C. *Discussion of Results*

As it can been observed in Table II, the SURF method in combination with FLANN, correctly compared the devices' images, as the top 5 final results of the known devices that had similarities with the unknown device (highlighted with grey), were of the same type with the unknown device (i.e. blood pressure monitors). More specifically, it can be observed that the iHealth Ease had 29.2% similarity with the TaiDoc TD-3128, iHealth Clear had 61.1% similarity with the TaiDoc TD-3128, Omron BP710 had 84.3% similarity with the TaiDoc TD-3128, iHealth Track had 90.7% similarity with the TaiDoc TD-3128, and Omron M3 IT had 97.9% similarity with the TaiDoc TD-3128. Even if the Omron M3 IT had the best overall similarity results, it is notable that it did not differentiate a lot with the next known device's similarity percentage (i.e. iHealth Track), since they had approximately a percentage of 7.2% difference rate, whereas even if it had a bigger number of difference rate among the overall similarity percentage of the other top 3 known devices, again the predicted device type was totally correct. As a result, it can be undeniably stated that the concept of computer vision can be effectively applied for identifying the type of devices of unknown nature.

On top of this, in order to verify the reliability and the correctness of the mechanism's result, this was compared with the corresponding result that was derived from the mechanism that we had proposed in [55]. More particularly, in this work, instead of using computer vision techniques, we had implemented classification techniques for predicting and recognizing the type of a device of unknown nature. In both cases, we had provided as an input the same unknown device (i.e. TaiDoc TD-3128), whereas in the prior work we used as a sample for our experiment 100 different known devices, whilst in the current work we used a sample of 30 different known devices. However, the number of the chosen known devices does not immediately affect any of the mechanisms' result, and thus it is not considered to influence the performance of them. Consequently, observing the results of the two (2) mechanisms, both correctly predicted that the type of the unknown device was a blood pressure monitor. Nevertheless, the current mechanism predicted that the unknown device was by 97.9% a blood pressure monitor, in contrast with the old mechanism that predicted that the unknown device was by 94.3% a blood pressure monitor, indicating that the current mechanism produced more precise and reliable results.

V. Conclusions

It is an undeniable fact that devices' recognition and integration is a very challenging research topic in the IoT area. On top of this, another continuously growing field is the one of the computer vision that can be easily adopted to cope with the aforementioned challenging topic of IoT devices. For that reason, in this paper we have conducted a study about the field of recognizing heterogeneous IoT devices of both known and unknown nature, by implementing computer vision techniques in order to identify their device nature (i.e. device type). For that purpose, we implemented a mechanism for facilitating the recognition of different IoT devices of unknown nature and the prediction of their devices' type, based on their images. In this mechanism, four (4) discrete stages were followed for recognizing an unknown device's type, by categorizing and mapping the unknown device's image with one or more images of known devices, and as a result finding out the device type of the unknown device.

Currently, we are working on the testing and the evaluation of the proposed mechanism by testing it with more heterogeneous IoT devices, of various domains. What is more, we aim to test our mechanism's efficiency by studying, implementing, and combining more computer vision techniques, regarding both feature detection and feature matching on the input images. Apart from this, our future work plans to update the mechanism by not only offering a Bluetooth interface, but also a general interface in which all the devices will be able to be connected regardless of their communication protocol. To be more specific, our primary goal is for our mechanism to be able to recognize and connect Wi-Fi enabled devices. Moreover, we are willing to extend our mechanism by finding a solution for coping even with the rarest cases that the mechanism is not able to find an image of an unknown device online in the Web. Finally, we aim to develop a security module for ensuring the originality of the devices' captured names and MAC addresses, avoiding potential spoof of them that could result into misleading decisions.

Acknowledgment

The research leading to this result has received funding from the European Union's Horizon 2020 research and innovation programmes under grant agreement No 826106 (InteropEHRate project) and grant agreement No 727560 (CrowdHEALTH project).

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," Computer networks, 54(15), pp. 2787–2805, 2010.

[2] "The internet of things is revolutionizing our lives, but standards are a must",https://www.theguardian.com/media-network/2015/mar/31/the-internet-of-things-is-revolutionising-our-lives-but-standards-are-a-must

[3] S. Jabbar, F. Ullah, S. Khalid, M. Khan, and K. Han, "Semantic interoperability in heterogeneous IoT infrastructure for healthcare," Wireless Communications and Mobile Computing, 2017.

[4] "Adding AI To The IoT", https://semiengineering.com/adding-ai-to-the-iot/

[5] "The future of IoT device management", https://www.networkworld.com/article/3258812/internet-of-things/the-future-of-iot-device-management.html

[6] "Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated", https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated

[7] "ARM predicts 1 trillion IoT devices by 2035 with new end-to-end platform", https://news.itu.int/arm-pelion-iot-end-to-end-platform/

[8] "Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025", https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[9] "5 Challenges Facing Health Care IoT in 2019", https://www.iotforall.com/5-challenges-facing-iot-healthcare-2019/

[10] A. M. Nagib, and H. S. Hamza, "SIGHTED: A Framework for Semantic Integration of Heterogeneous Sensor Data on the Internet of Things," Procedia CS, pp. 529–536, 2016.

[11] P. Gong, "Dynamic integration of biological data sources using the data concierge," Health Information Science and Systems, 1(1), 7, 2013.

[12] L. Martín, et al., "Ontology Based Integration of Distributed and Heterogeneous Data Sources in ACGT," HEALTHINF, pp. 301–306, 2008.

[13] M. M. Moazzami, G. Xing, D. Mashima, W. P. Chen, and U. Herberg, "SPOT: A smartphone-based platform to tackle heterogeneity in smart-home IoT systems," 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), pp. 514–519, 2016.

[14] P. F. Pires, E. Cavalcante, T. Barros, F. C. Delicato, T. Batista, and B. Costa, "A platform for integrating physical devices in the Internet of Things," Embedded and Ubiquitous Computing (EUC), IEEE, pp. 234–241, 2004.

[15] N. M. Ali,S. W. Jun, M. S. Karis, M. M. Ghazaly, and M. S. M. Aras, "Object classification and recognition using Bag-of-Words (BoW) model," 2016 IEEE 12th International Colloquium on Signal Processing & Its Applications (CSPA), pp. 216–220, 2016.

[16] X. Feng, Q. Li, H. Wang, and L. Sun, "Acquisitional rule-based engine for discovering Internet-of-Things devices," 27th Security Symposium (USENIX), pp. 327–341, 2018.

[17] C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," 1998 6th International Conference on Computer vision, pp. 555–562, 1998.

[18] "Understanding and Building an Object Detection Model from Scratch in Python", https://www.analyticsvidhya.com/blog/2018/06/understanding-building-object-detection-model-python/

[19] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2083–2090, 2013.

[20] "A Beginner's Guide to Object Detection", https://www.datacamp.com/community/tutorials/object-detection-guide

[21] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," Neurocomputing, 300, pp. 70–79, 2018.

[22] B. Gupta, A. Chaube, A. Negi, and U. Goel, "Study on Object Detection using Open CV-Python," International Journal of Computer Applications, 162(8), 2017.

[23] "What is image classification?", http://desktop.arcgis.com/en/arcmap/latest/extensions/spatial-analyst/image-classification/what-is-image-classification-.htm

[24] "Image Classification", http://www.sc.chula.ac.th/courseware/2309507/Lecture/remote18.htm

[25] "The Top 5 Uses of Image Recognition", https://imagga.com/blog/the-top-5-uses-of-image-recognition/

[26] "Image Classification using Deep Neural Networks — A beginner friendly approach using TensorFlow", https://medium.com/@tifa2up/image-classification-using-deep-neural-networks-a-beginner-friendly-approach-using-tensorflow-94b0a090ccd4

[27] P. Kamavisdar, S. Saluja, and S. Agrawal, "A survey on image classification approaches and techniques," International Journal of Advanced Research in Computer and Communication Engineering, 2(1), pp. 1005–1009, 2013.

[28] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using SIFT features and mean shift," Computer vision and image understanding, 113(3), pp. 345–352, 2009.

[29] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," European Conference on Computer Vision, Springer, pp. 404–417, 2016.

[30] L. Gómez, and D. Karatzas, "MSER-based real-time text detection and tracking," IEEE 22nd International Conference on Pattern Recognition (ICPR), pp. 3110–3115, 2014.

[31] M. Agrawal, K. Konolige, and M. R. Blas, "Censure: Center surround extremas for realtime feature detection and matching," European Conference on Computer Vision, Springer, pp. 102–115, 2008.

[32] R. Lienhart, and J. Maydt, "An extended set of haar-like features for rapid object detection," IEEE International Conference on Image Processing, Vol. 1, pp. I–I, 2002.

[33] S. Maji, and J. Malik, "Object detection using a max-margin hough transform," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1038–1045, 2009.

[34] Z. Lin, L. S. Davis, D. Doermann, and D. DeMenthon, D, "Hierarchical part-template matching for human detection and segmentation," IEEE 11th International Conference on Computer Vision (ICCV), pp. 1–8, 2007.

[35] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, 86(11), pp. 2278–2324, 1998.

[36] B. Zhang, Y. Gao, S. Zhao, and J. Liu, "Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor," IEEE transactions on image processing, 19(2), pp. 533–544, 2010.

[37] D. Forsyth, "Object detection with discriminatively trained part-based models," Computer, (2), pp. 6–7, 2014.

[38] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," IEEE conference on computer vision and pattern recognition, pp. 779–788, 2016.

[39] L. Li, T. Wang, and Y. Zhang, "Face detection using surf cascade," 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 2183–2190, 2011.

[40] E. Rublee, V. Rabaud, K. Konolige, K., and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," 2011 IEEE International conference on Computer Vision (ICCV), pp. 2564–2571, 2011.

[41] J. Knopp, M. Prasad, G. Willems, R. Timofte, R., and L. Van Gool, "Hough transform and 3D SURF for robust three dimensional classification," European Conference on Computer Vision, Springer, pp. 589–602, 2010.

[42] J. Li, and Y. Zhang, " Learning surf cascade for fast and accurate object detection," IEEE conference on computer vision and pattern recognition, pp. 3468–3475, 2013.

[43] J. W. Hsieh, L. C. Chen, and D. Y. Chen, "Symmetrical surf and its applications to vehicle detection and vehicle make and model recognition," IEEE Transactions on intelligent transportation systems, 15(1), pp. 6–20, 2014.

[44] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," Sensors, 12(9), pp. 11734–11753, 2012.

[45] "Connected Device: The Internet of Everything", https://www.bluetooth.com/markets/connected-device

[46] "Find MAC Address Vendors Now", https://macvendors.com/

[47] "Google Images", https://images.google.com/

[48] "OpenCV", https://opencv.org/

[49] "Introduction to SURF (Speeded-Up Robust Features)", https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html

[50] Dr. Mistry, and A. Banerjee, "Comparison of Feature Detection and Matching Approaches: SIFT and SURF," GRD Journals-Global Research and Development Journal for Engineering, 2, pp. 7–13, 2017.

[51] "Fast Approximate Nearest Neighbor Search", https://docs.opencv.org/2.4/modules/flann/doc/flann_fast_approximate_nearest_neighbor_search.html

[52] M. Muja, and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," IEEE Transactions on Pattern Analysis & Machine Intelligence, 11, pp. 2227–2240, 2014.

[53] J. S. Pan, Y. L. Qiao, and S. H. Sun, "A fast K nearest neighbors classification algorithm," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 87(4), pp. 961–963, 2004.

[54] "NetBeans IDE", https://netbeans.org/

[55] A. Mavrogiorgou, A. Kiourtis, and D. Kyriazis, "A comparative study of classification techniques for managing iot devices of common specifications," International Conference on the Economics of Grids, Clouds, Systems, and Services (GECON), Springer, pp. 67–77, 2017.