Personalized IoT Visitor Welcome System

Yunbo Chen, *Student, University of Washington Bothell*

Abstract—Internet of Things(IoT) has permeated the lives of many homeowners. ”Smart” things like smartwatches, smart cups are considered by more and more people as potential components of building a smarter home. In this project, We are proposing a home visitor management solution for homeowners to design IoT suite (lighting, photo gallery, music combination) for specific visitors. To be able to realize this system, we need to investigate the potential challenges that we would encounter during implementation. For example, face recognition, multimedia database and IoT network system. After careful consideration of the relevant approaches, we have come up with and implemented our system architecture using Amazon Web Service. Finally, we evaluate our system in terms of usability, latency, potential limitations and finally look towards its external validity in the future.

Index Terms—IoT, Cloud Computing, Edge Computing, Deep Learning, Face Recognition.

1 Introduction

THE advent of Internet of Things (IoT) has enriched our imagination on the possible ways in improving our lives. IoT extended the scope of computing into small devices and improved the availability and productivity in many fields such as medical treatment and field service [1] [2]. Nowadays, as the IoT devices get more and more pervasive and less costly, the use of IoT devices has been incorporated into home renovation where some essential home accessories are integrated with IoT technology and provide more interactive functionalities for homeowners.

Therefore, we would like to utilize the existing IoT devices and provide a new customized visitor management function for homeowners without them spending money on another new IoT devices. We would like to propose a home visitor tracking system where the homeowner could manage the visitors that visited the home and design a custom IoT suite, such as a combination of lighting, photos or music for the specific visitor, so that the next time this visitor comes in, the IoT suite customized for this visitor would be automatically activated.

For example, suppose Tom is the homeowner of his house. When Tom enters the home alone, he may want a pink light with a smart home assistant saying ”Welcome home, Tom” and then start an exciting rock song. However, if Tom is entering home with his friends, who are not in the Home Visitor Welcome System, the normal yellow light will be turned on, and the smart home assistant will only say ”Welcome, Tom and the new visitors” with some soothing music playing right after. It is an interesting project that would entertain both the homeowner and the visitors, at the same time providing an alternative safeguard to track visitors coming into the homeowner’s house, thereby improving the security of the home.

1.1 System Design

In terms of the home settings in this project, we are assuming a typical scenario for a typical homeowner. Beside some IoT devices such as smart photo frame, smart lighting and smart speakers, the homeowner requires a door sensor that could tell whether the door is opened or not. Then we need a camera that faces the door to take shots of the door when the door is opened.

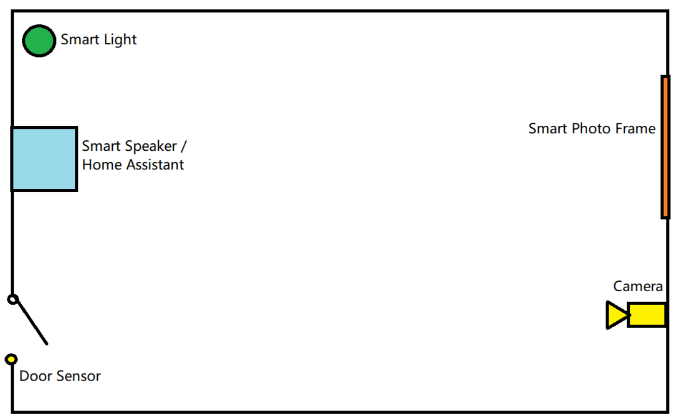

Fig. 1. Assumed home setting for the Home Visitor Management System.

Then in a typical workflow, the door sensor activates and sends an activation signal to the cloud, which alerts the camera to start recording (or take shots, we will discuss the feasibility of each of them in details later) visuals of what is happening at the door. Then the camera sends the footage into the cloud, which tries to catch faces appeared during the door opening and match the faces with the existing ones in the database. If there is a match, the cloud activates the corresponding IoT preset. The preset could be simply letting the speaker (smart home assistant) speak out a welcoming message including the guests’ names, or a combination of lighting, music, and photos previously set by the homeowner.

2 RELATED WORK

2.1 Face Recognition Techniques

To overcome some of the major technical challenges in this project, we would like to delve in research on state-of-the-art efficient face recognition approaches, and modern methods of facial data management. Besides, we would like to acquire some knowledge in edge computing [3] and fog computing [4], which may be in favor of this project because we need quick activation of the camera when the door sensor activates and sends data out to the cloud/edge.

The study of face recognition has been going on since the early 1970s when Kanade [5] tried to extract features from intensity-only images of human faces. As Jain [6] mentioned in his handbook on face recognition, the problem with feature-based recognition is that the features that the researcher deemed important are selected only by researchers themselves, which may not be entirely accurate and need parameter tuning. Even if the features are correctly selected and carefully measured to recognize faces, the system cannot be adjusted to be used in special conditions, namely, different exposure of lighting, contrast, background density of an image. Therefore the lack of flexibility makes it unlikely for us to use it on our system.

Another set of approaches for face recognition is to use statistical machine learning methods. Generally, researchers use Principal Component Analysis (PCA) [7] to process the images into "eigenfaces" in order to transform and reduce the dimensionality of the original images without losing much information. Therefore the processed image would highlight the key features for the images in the dataset. Then a statistical machine learning method is applied to the processed image. Li et al. [8] achieve an accuracy of 0.955 with 100 eigenfaces. However, Belhumeur et al. [9] find that PCA would fail to reduce undesired dimensions when there are multiple images existed in the dataset, which is very common to occur in our project because the face images of the homeowner will appear many times in the recorded images.

2.2 Face Recognition System

Our top-priority problem awaiting answers is an efficient face recognition approach in videos or consecutive images. In recent years, Folta et al. [10] registered a patent product assigned to Microsoft regarding face recognition and face data management in video content. They proposed a processing pipeline both for managing facial data and for retrieving face metadata, which is the object they store in the database. The face detection data (detected faces) is first matched against the face identification data in the face gallery. Then face tracking can detect additional faces in the frame. After getting all the face metadata, face grouping module will try to recursively divide the faces into groups of similar faces, which are also available for the user to access and help identify a person. The main advantage of this system is that it provides us with a comprehensive system for video face recognition. It contains the processing pipeline for getting face metadata as well as database management flowchart for constructing the system (see fig.4). Since it is an application for patent registration, it is feasible with a detailed blueprint on the face recognition system,

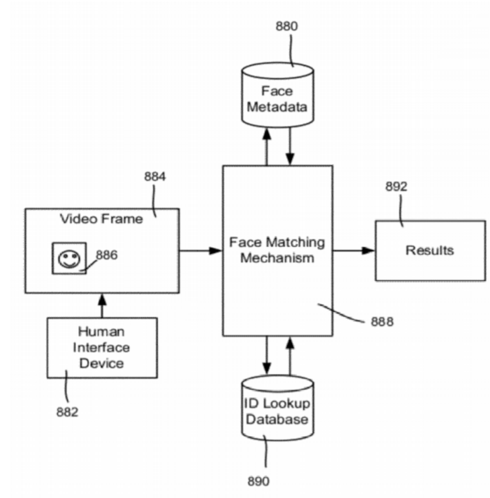

Fig. 2. Patented video face recognition system by Folta et al. [10].

key techniques and the hardware (the patent file contains desired hardware setting for the system as well). For the disadvantage of this system, we think that it is still impractical for us to implement because it is hiding the implementation of some critical parts of the workflow. For example, the system performs face detection on desired frames in the shot, whereas we do not know whether the frame is desired frame or not. It is related to other approaches that would distinguish key frames from the others and requires further research. Moreover, the system does not concern on the speed of face recognition and matching, while in our project, we do need the face matching to be as fast as possible.

3 PROBLEM STATEMENT

3.1 Possible Challenges

There are many challenges that we could think of pertaining to realizing this system. One of the biggest challenges is to detect and recognize faces from the footage of the door opening. We need to invest in whether we decide to use consecutive photo shots or direct video recording to capture the door opening moment. To make this decision, we should look at different approaches to face recognition in both videos and consecutive images. Another challenge, still on face recognition, is how to make the system memorize the recognized faces, which meanings storing the recognized face into the database, and be able to match it again from the database. From the best of our knowledge, we have seen the industrial-wise implementation of face recognition and face matching through the Photos app in Apple's iOS operating system, in which the app tries to extract all the faces appeared in user's photos and group the same faces together [11]. The user can go into People section and see the faces grouped by the app, and help it recognize more similar faces by confirming additional photos that the app finds similar to the confirmed faces but uncertain whether they are the same face or not. This technical feature is also available in Google Photos [12].

Beside face recognition and database management, there are also other small design problems that may not be as critical or technical as the challenges mentioned above but

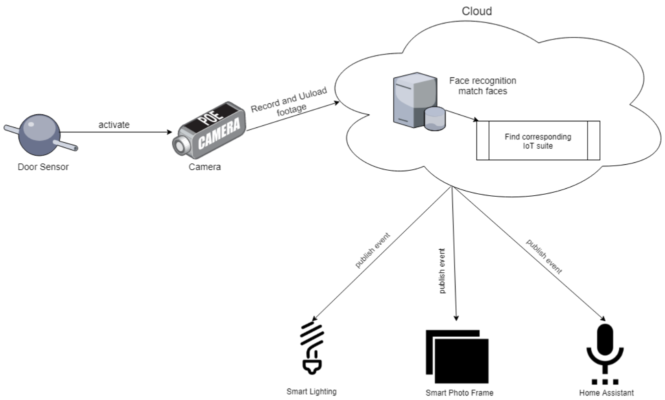

Fig. 3. Proposed abstracted system workflow of the Home Visitor Welcome System.

would affect the final demonstration. For example, how do we know if the person is coming into the house or going out of the house when the door opens? We can have two sensors A and B, A installed on the door, B installed outside the door. Then we can know from the order A and B are activated: if A is activated later than B, then the people are going out, and vice versa. Alternatively, we could train the cloud to distinguish whether the user is coming in or going out by analyzing the footage from the camera, which may save some cost in hardware but would take more training and computing time.

3.2 Abstraction of Problems

Our Home Visitor Welcoming System project can be divided or abstracted into three main problems/modules: a door-to-camera recording module, a face recognition database module, and an IoT suite activation module, as depicted in figure 3. The overall workflow between the modules is uni-directional. Therefore there is no exchange of information in between, which makes the system more feasible to implement and maintain overall.

For the door-to-camera module, we would like the door sensor to be the on-off switch for the camera so that the camera will start or stop recording instantly when the door is opened or closed. Another problem is, as described above, to distinguish whether people are coming in or going out of the house when the door is opened. The final decision on this design is related to the overall system architecture, and thus it can be figured out later in the implementation. To address this problem, we may need to look into different door activation triggers/sensors, as well as a commercial solution for home security camera mechanism.

For the face recognition database module, we are trying to solve the problem of making face recognition and storing face recognition data on the cloud. The module is also highly related to the solution of the previous problem/module since the approach to the face recognition is related to whether the camera is sending the video footage or just consecutive frame shots to the cloud. To address this problem, we would need to look into different mature face recognition cloud services, or machine learning cloud computing service in general.

For the last IoT suite activation module, we are trying to solve the problem of storage/retrieval of different IoT suites, as well as the activation of corresponding IoT devices. It is a practical problem since many types of smart furniture are controlled with different APIs and services. Some smart assistant devices are not compatible with IoT devices of different brands because of the competition between giant Internet companies. To address this problem, we need to look up activation ways of all different IoT devices to acquire a comprehensive understanding of essential ways of IoT things activation.

4 Design and Implementation

Based on the analysis on the problem, we have selected Amazon Web Service [13] [14] as the primary cloud service provider for this project. The detailed Amazon Web Service architecture is shown in figure 4.

The door sensor is simulated with an AWS IoT Button, which is a simple button that can send specific button payload to its own MQTT topic on AWS IoT Core and thus trigger relevant AWS Lambda function. The reason for choosing the IoT button is that it is simple to integrate with other AWS services, especially for triggering AWS Lambda functions.

The Home_Visitor_Tracker Lambda function, triggered by the door IoT Button, is deployed on the AWS IoT Greengrass Core device, which in our case is the camera device. In our implementation, we use AWS DeepLens as the core camera device. However, any portable single-board computer with camera device is feasible to execute this workflow. We choose AWS IoT Greengrass as the service to execute camera recording and the core of our architecture. One reason is that it is convenient to coordinate with other AWS services that are used in the rest of the workflow such as AWS S3, Rekognition, DynamoDB, etc. The other reason is that the use of edge computing, which is to offload some computing process to the edge device to reduce latency in execution. The Home_Visitor_Tracker Lambda function is the primary function that coordinate with other AWS services. It mainly consists of 4 parts: face detection, recognition, IoT suite retrieval, and IoT suite activation.

4.1 Face Detection

The first part of the Lambda function tries to consume the video and return a set of different faces occurred in the video. The face detection technology is supported by an external library called Face-Recognition by Adam Geitgey [15]. The library contains a pre-trained face recognition model that could efficiently detect and recognize faces with high accuracy. It also uses a face_encoding object to encode and store the detected face, which is convenient for our use case.

At the beginning of each camera activation, which is also the start of the lambda function activation, the function will create two empty lists, one to store face_encoding objects, the other to store the face shots image files for later upload. On each frame, the device will detect the faces appeared in that frame, encode each face as a face_encoding object, and then compare each face with the face_encoding objects in the

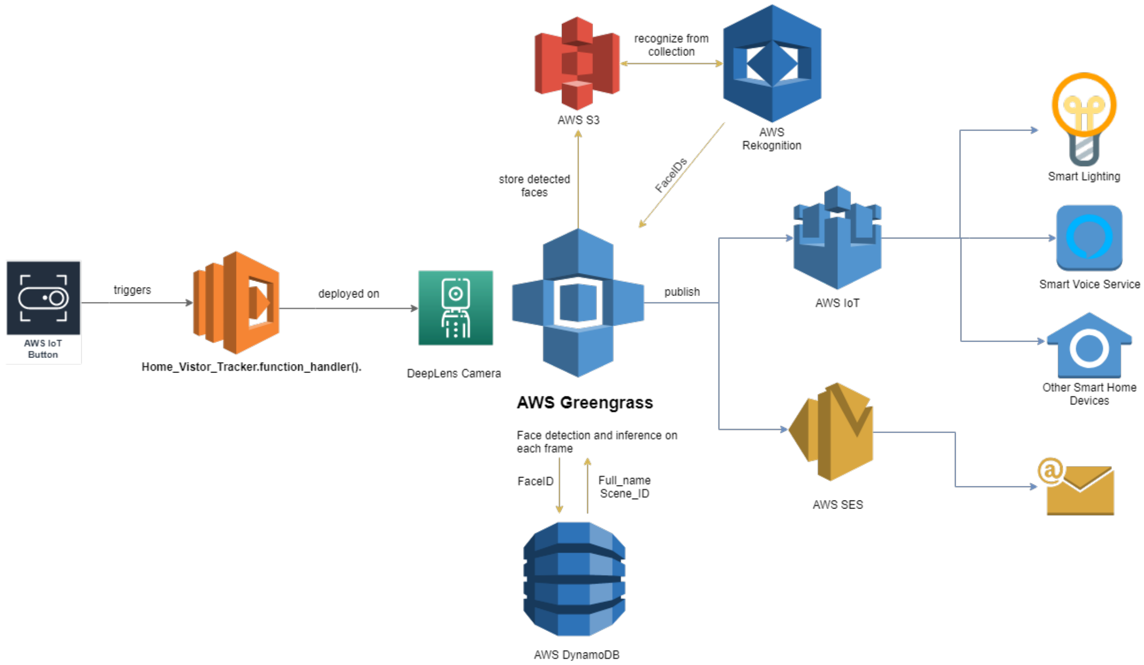

Fig. 4. Proposed detailed system workflow of the Home Visitor Welcome System.

list. If the face is not in the list, indicating that the face is not previously appeared in the video, the face_encoding object will be appended onto the list, while its face shot image will be stored in the other list. After processing all the frames in the video, the face shots list will be propagated with all different faces appeared in the video.

The function then will upload all the face shot images onto AWS S3 bucket with the same timestamp prefix, so that all the different faces detected in this video session will be stored in the same "folder," which will be easier for AWS Rekognition to find the images later on. The uploaded images are also set with the permission of "public-read" so that the homeowner can view them on any devices later in the email.

4.2 Recognition

The next part of the lambda function is to use AWS Rekognition to recognize the faces appeared in this video session. To do this, the homeowner needs to create a Rekognition collection beforehand. Since AWS Rekognition does not have a graphic user interface or console interface to operate, the user has to use AWS-CLI to configure the AWS Rekognition service. There are three simple steps: first is to create a collection; second is to upload the images of the faces that the user would like our system to recognize; third is to do index-face() command to let the Rekognition index those faces and give a "faceID" for each of the faces.

In the Home_Visitor_Tracker lambda function, the function will be directed to the images on AWS S3 with the timestamp prefix and search familiar faces in each of them. If there is a match, it will return the response with the matched "faceID" for us to retrieve corresponding IoT suite.

4.3 IoT Suite Retrieval

After recognizing who has entered the door and retrieved the "faceID" from AWS Rekognition, the next part of the lambda function will then go to AWS DynamoDB to retrieve the corresponding visitor full name and IoT suite.

The AWS DynamoDB contains an IoT suite table with the primary key of "faceID" and other keys of "Full Name," IoT suite "Scene ID" and "Priority" values. The "Full Name" value is used to fill the email template with visitor names. The "Scene ID" value is used to activate the LIFX lighting scene preset by the homeowner. The "Priority" value is used to retrieve the importance of the visitor and determine whose IoT suite will be selected to activate when there are more than one known faces recognized in the video.

In the Home_Visitor_Tracker lambda function, the function will first check the parity between the number of known faces and the number of different faces detected. If they are not equal, meaning that there are visitors that are not in the system, the system will directly activate visitor IoT suite. If the numbers are equal, the function will go to DynamoDB with the "faceID"s retrieved from Rekognition and retrieve the full names, IoT suite IDs and priority values of those known faces, and then compare the priority values of the known faces and select the most prioritized "faceID" to activate its corresponding IoT suite.

4.4 IoT Suite Activation

After selecting which IoT suite to activate, the function will then parse the JSON object stored in the scene ID and activate the IoT suite via different APIs determined by different IoT devices. In our implementation, we only have a LIFX lighting device, which is activated by scene ID through the HTTP request. Therefore we wrap the scene ID in the payload, along with authentication token in the header, and send the HTTP request to activate the IoT suite.

TABLE 1
Results of Scenario Experiment

Scenario	Distance	Lighting	Correctness	Response Time
1	5	Green	Yes	3.05
2	5	Yellow	Yes	2.01
3	5	Yellow	Yes	2.31
1	8	Green	Yes	3.16
2	8	Yellow	Yes	2.36
3	8	Yellow	Yes	2.29
1	10	Pink	No	2.34
2	10	Yellow	Yes	2.25
3	11	Off	No	N/A
1	11	Off	No	N/A

The full name key will be used to publish an email to the homeowner with the content of the full names of known visitors, along with links to the face shots stored in the S3 bucket. The email is sent with AWS SES service.

5 Evaluation

To evaluate our design, we have selected the metrics of accuracy, latency, user experience and cost. We have set up three scenarios for the experiment. The first scenario is that only the known faces are entering the house. The second scenario is that a known face with another visitor is entering the house. The third scenario is that only the visitor is entering the house. There are two ”faceIDs” stored in the AWS DynamoDB. One is Robert with a priority of 1 and an IoT scene ID of 1 with pink lighting. The second face is Winnie with a priority of 2 and an IoT scene ID of 2 with green lighting. For general visitor IoT scene, it has an ID of 3 with yellow lighting. Typically scenario three should not happen since the visitor should not have an access key of the house.

For the detailed metrics, the response time is the amount of time it takes for the system light up corresponding IoT lighting after the camera recording is complete. Correctness is whether the activated IoT lighting is correct.

From the result shown in Table 1, we can see that the system performs well with a distance under 8 meters. When the distance goes to 10 meters, the error occurs in scenario 1. After checking the email, we have found that the system only detects Robert’s face during the camera recording since Robert was a bit closer to the camera than Winnie. When the distance goes beyond 10 meters, the camera failed to detect any faces in the video session. This failure of detection in a greater range is caused by the down-scaling that happened before doing face detection on each frame. We purposefully down-scale each frame before face detection in order to increase the number of the frame processed in the door opening duration.

In terms of the response time, we can see that scenario one typically takes a more protracted process time than scenario 2 and 3. It is reasonable because AWS Rekognition needs extra time in retrieving ”faceID” for known faces. More importantly, since there are no visitors in scenario 1, the system needs to perform retrieval from AWS DynamoDB and then run a priority algorithm to figure out whose IoT lighting to perform. Therefore, entering home with visitors will typically take less time for the IoT suite to be activated, which is in favor of the design purpose since visitors would not feel the ”welcome”.

In terms of the user experience, the system does require some basic programming skill to use AWS-CLI to set up Rekognition collection and index faces. Other than that, the system does not require the user to actively participate or interact with it other than pressing IoT Button at the beginning. Therefore, the overall user experience is smooth.

In terms of the cost on the system, a typical Raspberry Pi with camera V2 module costs $65.23 [16]. We tried to compare the cost with a commercial smart home system with the same functionality, but we did not find any system that has implemented the same face-recognition IoT suite activation functionality as we did in our system. Nonetheless, a commercial home camera with smart lighting would cost more than $200 [17]. In this case, even if we add the cost of using different AWS services, our system is cheaper than the commercial home cameras.

6 Conclusion and Future Work

In the paper, we have proposed and implemented a Home Visitor Welcome System that can activate different IoT suites (lighting, voice message, music, etc.) preset by the homeowner according to different visitors entering the home. We designed and implemented our system using AWS [13] [14] services and evaluate our system in terms of accuracy, latency, user experience, and cost. The system can accurately recognize the visitors and activate corresponding IoT lighting when the camera is within 8 meters from the door. Despite the unstable waiting time of the IoT Button, the system is able to activate the suite within 3 seconds after the recording is done, which is acceptable for users. In terms of overall user experience, despite small configuration that requires command-line coding skill, the overall system does not require active interaction with the system. Overall, the system is a well-designed working system with costs lower than any of the commercial camera system, not to mention its uniqueness in the functionality.

However, there are limitations pending improvements. The precision of the camera goes off when the distance between the user and the camera goes over 10 meters due to the down-scaling frame processing algorithm. The IoT Button is unstable in terms of response time, which could significantly affect user experience. The door closing trigger is still not implemented since IoT Button takes too long to trigger the Lambda function. Moreover, the currently supported IoT suite only includes the LIFX lighting, which can be activated using the HTTP request.

In the future, we will look into faster ways to replace the IoT Button and trigger the lambda function inside the AWS Greengrass core device instantly and directly, such as Bluetooth or other near-field communication approaches. We also will create an IoT suite object to store different IoT things in the house and the corresponding APIs on activating the suite so that we could manage the activation of different IoT suite by using our own parser function to read the object.

REFERENCES

[1] Econsultancy. 10 examples of the internet of things in healthcare econsultancy, Feb 2019.
[2] The top 5 ways iot is revolutionizing field service.
[3] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.
[4] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.
[5] Takeo Kanade. Picture processing system by computer complex and recognition of human faces. 1974.
[6] Anil K Jain and Stan Z Li. *Handbook of face recognition*. Springer, 2011.
[7] Ian Jolliffe. Principal component analysis. In *International encyclopedia of statistical science*, pages 1094–1096. Springer, 2011.
[8] Huaqing Li, Shaoyu Wang, and Feihu Qi. Automatic face recognition by support vector machines. In *International Workshop on Combinatorial Image Analysis*, pages 716–725. Springer, 2004.
[9] Peter N Belhumeur, João P Hespanha, and David J Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. Technical report, Yale University New Haven United States, 1997.
[10] Florin O Folta, Yaming He, King Wei Hor, Minesh G Shilotri, Stacey Spears, and Chuang Gu. Face recognition in video content, July 23 2013. US Patent 8,494,231.
[11] About people in photos on your iphone, ipad, or ipod touch, Oct 2018.
[12] Label your face group - iphone i& ipad.
[13] Jörg Polzehl and Vladimir Spokoiny. Propagation-separation approach for local likelihood estimation. *Probab. Theory Related Fields*, 135(3):335–362, 2006.
[14] Jörg Polzehl, Kostas Papafitsoros, and Karsten Tabelow. Patchwise adaptive weights smoothing. Technical Report 2520, WIAS Berlin, 2018.
[15] Adam Geitgey. ageitgey/face_recognition, Mar 2019.
[16] Raspberry pi camera module v2-8 megapixel,1080p.
[17] Smart indoor security camera - netatmo welcome.