IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-11, Nov 2019 E-ISSN: 2347-2693

Voice Enabled Smart Home Assistant for Elderly

Sujitha Perumal[1]*, Mohammed Saqib Javid[2]

[1,2]Dept. of Computer Science, Dr. Mahalingam College of Engineering & Technology, Tamil Nadu, India

*Corresponding Author: sujithaperumal3@gmail.com, Tel.: +91-9442582602

DOI: https://doi.org/10.26438/ijcse/v7i11.3037 | Available online at: www.ijcseonline.org

Accepted: 16/Nov/2019, Published: 30/Nov/2019

***Abstract*:** In this paper we present our project that focuses on developing a Voice enabled Smart Home Assistant that will help find items in a home and also support natural and secure interaction using voice recognition.The software will be helpful for elderly people and those with Alzheimer's disease to help find items at home. It can also be used by people who often forget where they keep their things as well as large organizations to manage complex inventory, without the friction and multiple steps of keeping inventory updated. The project will be relevant to society, as users will be able to communicate with the Smart Home Assistant in a natural way and security will also be ensured as no unknown users will have access to the software. The software can also be accessed through mobile application.The Voice enabled Smart Home Assistant will employ natural language processing techniques to understand the user's request to identify the location of the object and report it to the user. The software will be integrated with Amazon Alexa which is a cloud-based voice service and the users can access the software using the Amazon Echo device.

Keywords: Personal Assistant, Natural Language Processing, Voice Recognition

I. INTRODUCTION

Today, virtual personal assistants, which can essentially organize large parts of our lives, are a standard feature on smartphones worldwide. Amazon Alexa is one such virtual personal assistant.Alexa is Amazon's cloud-based voice service available on tens of millions of devices from Amazon and third-party device manufacturers. With Alexa, natural voice experiences can be built that offer customers a more intuitive way to interact with the technology they use every day.The personal assistant 'Alexa' is used for the interaction module with the users, as it is more feasible and supports natural language processing and also helps users interact with the device naturally and use the service in an efficient way. The main module of the software is the natural language processing implementation which enables the natural interaction between the user and the personal assistant. Another main module of this software is the video processing implementation. This module processes the objects present in real time, identifies them and then stores the objects in a database.A dynamo database which is a service of Amazon Web Services is used to store the items and their locations. The users can add or remove items from the database. The database will be updated periodically based on the inputs received from the Video processing module as well as voice interaction with the users. The location of the object will be retrieved from the database and the response will be announced to the user. The main objective of this project is to develop a Voice enabled Smart Home Assistant that will help people find items/objects in a home environment using Natural Language processing techniques and video processing techniques to locate objects of interest.

Rest of the paper is organized as follows - Section I contains the introduction, Section II contains literature survey, Section III contains methodology, Section IV contains modules, section V describes results and discussion, Section VI contains conclusion and future work.

II. LITERATURE SURVEY

1.Human Computer Interaction Using Personal Assistant: Hyunji Chung et al. proposed that assisting users in performing their tasks is an important issue in Human Computer Interaction research. A solution to deal with this challenge is to build a personal assistant agent capable of discovering the user's habits, abilities, preferences, goals and accurately anticipating the user's intentions [1]. In order to solve this problem in an intelligent manner, they proposed that the assistant agent has to continuously improve its behavior based on previous experiences. By endowing the agent with the learning capability, it will be able to adapt itself to the user's behavior.

Abhay Dekate et al. stated that two of the most important issues that personal assistant agents have to deal with are learning and adapting to the user's preferences. In order to solve the problem of user's assistance in an intelligent manner, the assistant agent has to continuously improve its behavior based on the experience of the actions taken by

users that successfully achieved a specific task [2]. In this direction, the agent has to be endowed with the learning capability, thus becoming able to adapt itself to its dynamic environment. Considering that the agent usually performs a substantial number of repetitive tasks, previous experiences can be used to handle similar future situations.

Veton Kepuska proposed two main challenges in the area of human-computer interaction. First, how to know the user, and second, how to assist the user. They proposed that most of the previous work has contributed to the first challenge by improving the interaction between users' and PA agents, learning users' preferences and goals, providing help at the right time and so on. As Chen and Barth have mentioned, only knowing the user is not enough for solving all problems, as a good PA agent is supposed to have some knowledge related to the tasks to be done. Problems can be solved in an intelligent way by reasoning, making inferences and learning [3].
Thus, it is inferred that for a better human computer interaction having an Intelligent Personal Assistant improves the efficiency. The personal assistant used must be able to learn and supervise by itself. The main reason for the usage of the personal assistant is that it will be capable to interact with users in a natural way.

2.Object Detection Framework for High Performance Video Analytics: Ashiq Anjum says that object detection and classification are the basic tasks in video analytics and is the starting point for other complex applications. He proposed that the analysis criteria defines parameters for detecting objects of interests (face, car, van or truck) and size/color based classification of the detected objects. The recorded video streams are then automatically fetched from the cloud storage, decoded and analyzed on cloud resources. The operator is notified after completion of the video analysis and the analysis results can be accessed from the cloud storage. The Video Stream Acquisition component captures video streams from the monitoring cameras and transmits to the requesting clients for relaying in a control room and/or for storing these video streams in the cloud data center. The video analysis approach detects objects of interest from the recorded video streams and classifies the detected objects according to their distinctive properties. [4].
Thus, it is inferred that object recognition can be used by streaming videos than collecting data from images. This increases the accuracy and makes it easy to retrieve the details of the object when asked by the user.

III. METHODOLOGY

A. Existing System

The existing system as shown in Figure 1 uses Echo device in which the personal assistant 'Alexa' is incorporated. Using the echo device, users can access basic in-built skills such as 'play music', 'what is the weather?' etc. This is achieved when the user calls Alexa using the wake word 'Alexa' and then calling the appropriate command.

For example, when a user wants to know about the weather, he could say 'Alexa, what is the weather?' Here, 'What is the weather?' is the speak command which is sent to the echo device. The device then calls the task manager via controller and converts this speech to text. This is then sent to the service manager which identifies the command and sends it to web service adapter. This adapter analyzes the command and if a match occurs, it sends an appropriate request to the firebase cloud server. The firebase cloud server then calls the appropriate intent and runs the corresponding code. The response is sent back to service manager which converts the text to speech. The output speech prompt is sent to device via controller and task manager. The device outputs the speech prompt to the user.

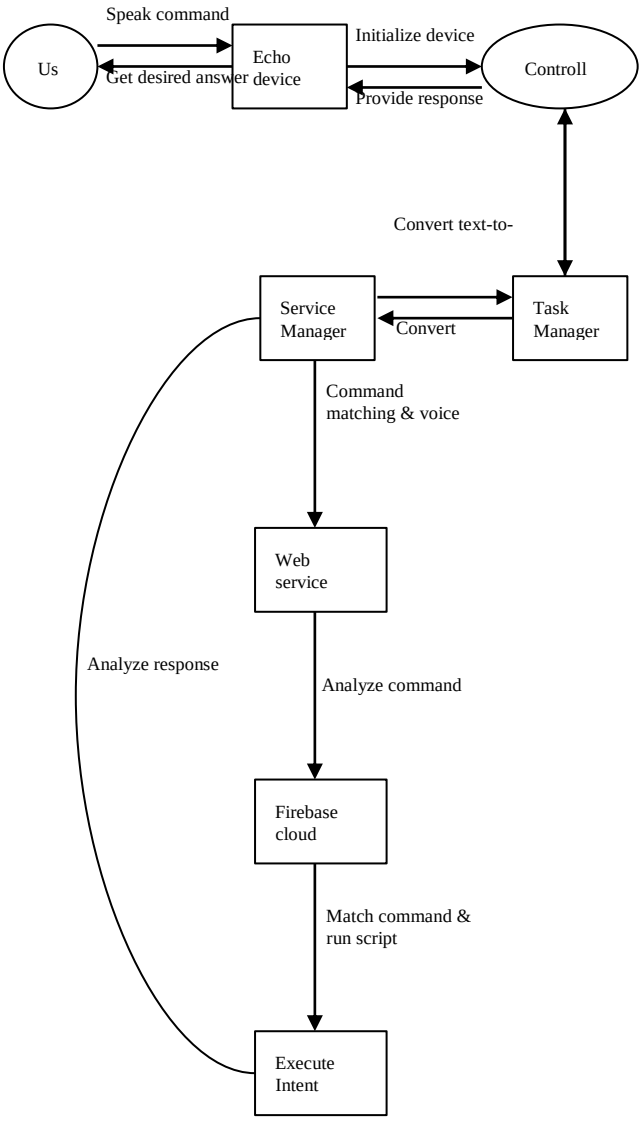

Figure 1: Block diagram for Existing System

The communication between the user and personal assistant occurs as follows:

- The user speaks to Echo, using trigger words so that Echo knows that it is being addressed, and identifies the Skill that the user wishes to interact with. For example, "Alexa, ask object finder where is the watch". In this case, "Alexa" is the trigger word to make the Echo listen, and "object finder" identifies the skill that the user wants to direct their enquiry to.
- Echo sends the request to the Alexa Service Platform, which handles speech recognition, turning the user's speech into tokens identifying the "intent" and any associated contextual parameters. In the example, the "intent" would be that the user wants to know about the place of the watch, and the context for that would be that they are interested specifically in the place of the watch. Intents and possible parameter values for them are held by the Alexa Service Platform as configuration items for the Skill.
- The intent and parameters for the user's request are then sent as a JSON encoded text document to the server side Skill implementation for processing. The Alexa Service Platform knows where to send these requests as it maintains a set of backend URLs or Lambda ARNs for each Custom Skill.
- The Custom Skill receives the JSON via an HTTPs request (or if the Custom Skill is implemented as an AWS Lambda function, via invocation of the Lambda function at the configured ARN).
- The Custom Skill code parses the JSON, reading the intent and context, and then performs suitable processing to retrieve data appropriate to those. A response JSON document is then sent back to the Alexa Service Platform containing both the text that Alexa should speak to the user and markup containing text and an optional image URL for a "card" that appears in the Alexa companion smart phone app.

The Alexa Service Platform receives the response, and uses text to speech to speak the response to the user while also pushing a card to the companion app.

B. Proposed System

In the proposed system, Voice enabled Personal Assistant as shown in Figure 2 will employ natural language processing techniques to understand the user's request and then use image recognition to identify the location of the object and report it to the user. The software will be integrated with Amazon Alexa which is a cloud-based voice service and the users can access the software using the Amazon Echo device.

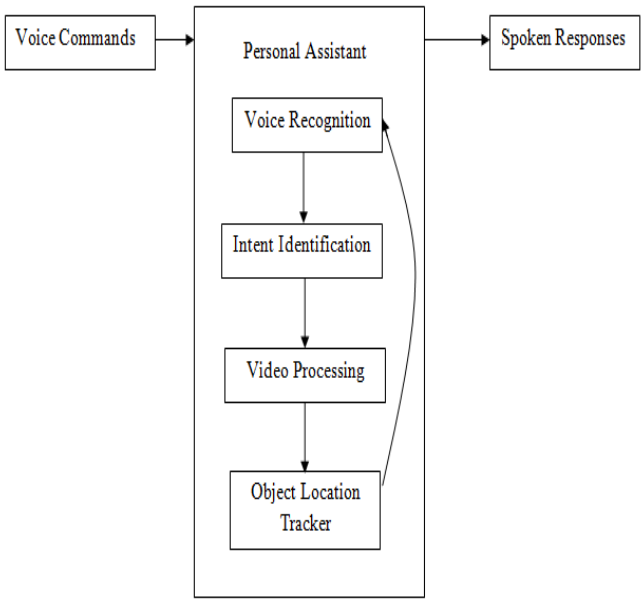

Figure 2: Block diagram for Proposed System

The whole dataset is divided into training dataset and test dataset (i.e.80% and 20% respectively). Fuzzy domains and regions are generated for each attribute in the dataset. The minimum and maximum value of an attribute forms the domain. By using triangular membership graph and based on the domain it is divided into three regions namely low, high and medium.

IV. MODULES

The proposed system has the following modules:

- Speech Recognition
- Intent Processing
- Video Processing

Speech Recognition

Initially, when the user speaks the wake word, the Amazon Echo device gets started. The wake word for an Echo device consists of 'Alexa' and 'Echo'. Users can choose their desired wake word that starts the Echo device and calls the in-built personal assistant-Alexa. The next step is to ask Alexa to open a particular skill and perform the necessary actions given as an input by the users. For example, in order to open the object tracker skill, the command "Alexa, open object tracker" is used. So when the user utters this sentence, the personal assistant recognizes that the user is trying to access the object tracker skill and calls it.

Once the skill is called, the personal assistant recognizes the utterance spoken by the user, converts it into text, processes the text to gather information about the intent, identifies the appropriate intent and calls it. The intent refers to the

function that executes a particular task. The intent then executes the necessary functions as requested by the user and processes the data. The processing of data occurs on the server side.

After the data is processed, the response needs to be sent to the user. For this to happen, the processed data is converted from text to speech and then it is sent to the personal assistant. The personal assistant-Alexa then tells the response to the user. Figure 3 shows how the speech recognition works.

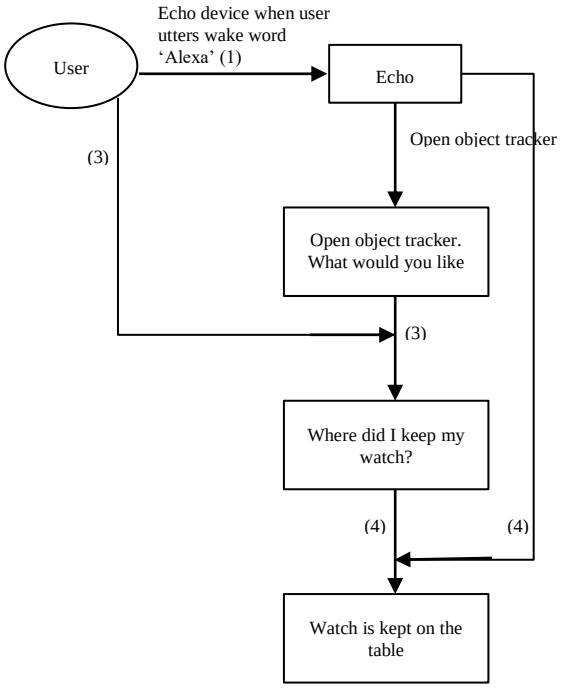

Figure 3: Flow Diagram for speech recognition

For example, if the user wants to know about the location of his/her watch, the user has to call the skill first. This is done by speaking the command "Alexa, open object tracker". The skill is called and it starts with a welcome message to the user. So, now when the user utters the question "Where did I keep my watch?" the skill initially recognizes the words spoken by the user. Then on processing the words, it spots the keywords 'Where' and 'watch' and identifies the intent as "FindObjectIntent". The skill then calls the intent to perform its operation/task (i.e.) to search the database whether an object of the same name 'watch' is found or not. If found, the location of the watch is retrieved from the database and sent to the personal assistant. The personal assistant then converts the text into speech and utters the response (i.e.) the location of watch to the user.

Generally, all skills developed for Alexa consists of a set of pre-defined commands using which the skills can be accessed. This aspect requires users to memorize the commands of skills that they have enabled for their device. Memorizing a large number of commands is tedious and less user-friendly for the users.

For a skill with no Natural Language Processing implemented, it only accepts commands that are specifically written for them and executes the necessary tasks and not other commands although they are similar in meaning, whereas, for a skill with Natural Language Processing implemented, it accepts all commands that are similar in meaning and executes the necessary tasks. This way, skills become more interactive and user friendly.

Therefore, in order to make skills interactive and user-friendly, NLP has been implemented in this system. The implementation involves capturing the keywords and comparing it with a set of equivalent list of words. If the keyword is found to be similar with any one of the equivalent set of words, then the appropriate intent is called and the functions are executed. This way, users need not memorize all the commands needed to execute a particular skill. Instead, they can speak in natural language and the skill still responds to them with appropriate answers by identifying the keywords.

For example, let a skill that has no NLP implemented, consists of the following commands:

- "get an object"
- "add an object"
- "delete an object"

When the user utters the sentence "add an object", the skill recognizes the words and identifies that the user wants to add an object to the database. Whereas, when the user utters the sentence "store my object", the skill does not recognize the words and fails to execute it although the keywords 'add' and 'store' imply the same meaning.

When the same skill is implemented with NLP, it understands that both the keywords 'add' and 'store' imply the same meaning and therefore calls the intent that adds an object to the database. This way, user interaction and user experience has been improved in this system developed.

Intent Processing

After the intent is identified, the next step is to process the data given as an input from the user and perform the necessary actions. Slots, also called as keywords/values are gathered from the user input and processed.

For example, when users say "Store my watch on the table", the skill identifies the intent as 'AddObjectIntent' by spotting the slot value 'store'. So, the 'AddObjectIntent' is called. The intent then captures the slot values 'watch' and 'cupboard' in 'ObjectName' and 'ObjectLocation' slot

names respectively. These slot values are then sent to the lambda function (i.e.) the code that processes the data.

The work of this lambda function is to store the object name and object location in the database so that when the user asks for the location of the object later, the skill retrieves the location of the object to the user. Before the object name and object location are stored in the database, the lambda function checks whether the object already exists in the database.

In this case, the object 'watch' is initially checked in the database to confirm that it doesn't exist already. If it exists, the object will not be added and the response "Watch is found on the table" will be spoken to the user. On the other hand, if the object does not exist, then, the slots 'watch' and 'table' will be added to the database.

The database used in this system is called as 'DynamoDB', a service provided by the Amazon Web Services. The database consists of the following attributes:

- Name - Name defines the object name which in our case is 'watch'.
- UserId - UserId is associated with the users' account. It is unique for every individual user. Each user connects to the Echo device with the Amazon Alexa mobile application. As each user connects to the Echo device, a userId is created. The skill captures this UserId indirectly by requesting the Echo device. This attribute is used in this system because when different users use this skill, the skill has to differentiate the users.
- Location – Location defines the object location which in our case is 'table'.
- Time – Time refers to the time when the user has told the skill to store a particular object in a particular location. This data is also got indirectly with the help of lambda function. Time zone is set in the lambda function. This attribute helps users know when they last added an object to the database.

The database looks similar to Table 1.

Table 1: DynamoDB Database

ObjectName	Name	Location	Time	UserID
Watch	Sujitha	Shelf	01.37 pm	amzn1.ask.account.AH3BH
Pen	Gita	Table	11.28 am	amzn1.ask.account.ANC4l

Another major aspect of this module is that lambda function is able to recognize words that are similar in meaning (i.e.) for example, the lambda function recognizes that both 'watch' and 'wristwatch' mean the same.

For example, the user calls the 'Object Tracker' skill and asks it to store the object, say, 'watch' in database with the location as 'table'. As shown in Figure 4, the skill checks whether the object is already present and stores it in the database.

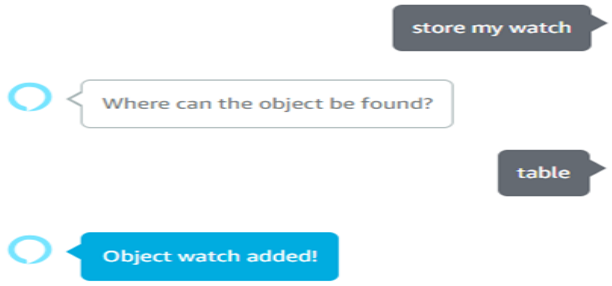

Figure 4: Storing an object in the database

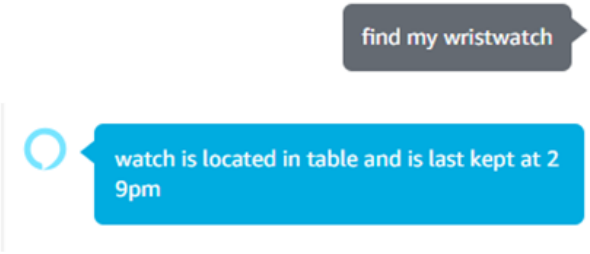

Figure 5: Retrieving an object from the database

Later, when the user calls the skill and asks for the location of the object 'Wristwatch', the skill recognizes that both 'watch' and 'wristwatch' mean the same and the user is trying to find the location of the object 'watch'. Therefore, as shown in Figure 5, the skill responds to the user with the location of object 'watch' to the user which in our case is 'table'.

Video Processing

One way of storing the objects in the database is by giving information to the skill about the object name and object location. The main objective of this system is to find objects for users without them telling any information to the skill. This method is tedious and makes users frustrating to give information every single time.

An alternative approach to this method is video processing. This method is implemented with the help of web cameras. It helps identify objects in real time using web cameras and AWS tool called 'Rekognition'.

Initially, using AWS Kinesis Video Stream service, a video stream is created. Amazon Kinesis Video Streams makes it easy to securely stream video from connected devices, in our case, web cameras, to AWS for processing data. After

creating the video stream, the next step is to connect the web cameras to the stream so that the video starts streaming. This can be achieved using the software Msys2. The video is streamed using a plugin called 'Gstreamer' plugin.In the Msys2 software, initially, the AWS credentials are setup and then the video is streamed to the Kinesis Video Stream by providing the stream name and AWS access key and AWS secret key.

Next step is to process the streaming video data and detect objects. This is done using the Single Shot Multibox Detector(SSD) algorithm. The algorithm is coded in a lambda function and gets linked to the video stream. So, when the video stream starts streaming, the lambda function starts detecting the objects present in the video. The lambda function then loads the detected objects to the 'ObjectsTracker' database. So now, when the user asks the skill where a particular object is, the skill searches the database and retrieves the location to the user.

For example, when a user searches for an object, say watch, which is not stored in the database, initially the intent 'GetObjectIntent' will be called and the object 'watch' will be searched in 'Objects' database. If there is no match, then the web cameras that are fixed at common places in a home and are integrated with this skill will be launched and asked to scan objects kept in those places. The objects are detected and they are converted into text and stored in the database and then compared with the keyword 'watch'. If any of those match with watch, then the location of the watch is returned to the skill which then sends the response back to the user.

V. RESULTS AND DISCUSSION

All the modules of the proposed system have been developed and tested for different scenarios. Performance evaluation has also been carried out.

The results of the tested modules are listed below:

A. Launch Request

The 'Object Tracker' skill is launched upon the invocation of launch request. In this case, "open object tracker" launches the skill and sends a welcome response to the user as shown in Figure 6.

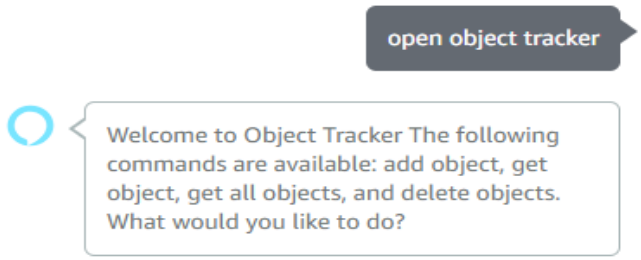

Figure 6: Execution of Launch Request

B. Add Object Intent

When a user says "open object tracker and add my watch to the cupboard", 'Object Tracker' skill is launched and the intent 'AddObjectIntent' is called on recognizing the keyword 'add'. This intent then collects the values 'watch' and 'cupboard' in the slots 'ObjectName' and 'ObjectLocation' respectively sends it to the 'Objects' database where these values are verified whether they already exist or not. Depending on this, they will either be added to the database or not. If they are added, the user will be intimated as "Watch has been added to the cupboard" as shown in Figure 7.

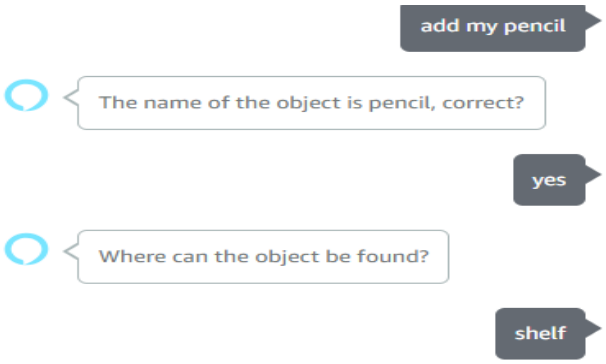

Figure 7: Execution of Add Object Intent

C. Delete Object Intent

If the user asks the skill to "Retrieve my watch", the intent 'DeleteObjectIntent' is called on recognizing the keyword 'retrieve'. This intent collects the value 'watch' in the slot 'ObjectName' and sends it to the 'Objects' database where the value is compared with all other values in the database. If a match appears, then that object is removed from the database. The response "Watch has been removed" is sent back to the user as shown in Figure 8.

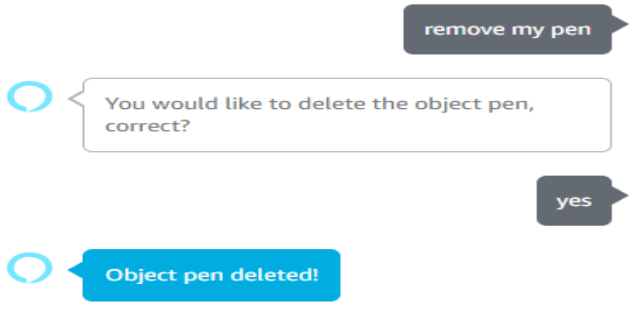

Figure 8: Execution of Delete Object Intent

D. Get Object Intent

If the request given is "where did I keep my watch?", the intent 'GetObjectIntent' is called on recognizing the keyword 'where'. It then collects the value 'watch' in the slot name 'ObjectName' and sends it to the 'Objects' database. The database checks this value with all other names in it and if a

match appears it returns back the location of the watch to the user like "The watch is in the cupboard and is last kept at 9.15pm". If a match does not appear, then the 'ObjectsLocation' database is called and the name of the object given by the user is compared with the values in the database. If there is a match, then the possible locations are sent back as response which is intimated to the user as "watch cannot be found. But the possible locations where it could be kept are cupboard, shelf and table" as shown in figure 9.

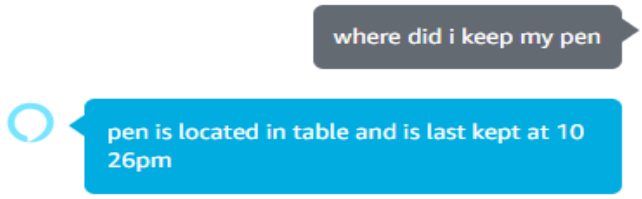

Figure 9: Execution of Get Object Intent

- **PERFORMANCE EVALUATION**

The performance of the developed system is assessed in terms of object detection accuracy and utterance recognition accuracy. The Alexa skill has been developed and published in Alexa Skills Store and is accessible to users all over the world.

$$Accuracy\ of\ object\ detection\ = \frac{Number\ of\ objects\ detected}{Total\ number\ of\ attempts} \quad (1)$$

$$Utterance\ detection\ accuracy = \frac{Number\ of\ utterances\ recognized}{Total\ number\ of\ utterances} \quad (2)$$

The comparison based on the accuracy between the objects detected without using video processing and objects detected with video processing is shown in Table 2. The accuracy obtained in detection of objects using video processing is greater than the accuracy obtained in detection of objects without using video processing.

Table 2: Comparison based on the accuracy of object detection

Objects	Existing System Without Video Processing	Developed System with Video Processing
Cellphone	No	Yes
Laptop	No	Yes
Book	Yes	Yes
Clock	Yes	Yes
Bottle	No	No
Total Objects Detected	2	4
Accuracy	20%	80%

Table 3 shows the comparison based on the utterances recognized by the system without NLP and the utterances recognized by the system with NLP. The number of utterances recognized by the system with using NLP is greater than the number of utterances recognized by the system without using NLP.

Table 3: Comparison based on the accuracy of utterances recognized

Utterances	Without NLP	With NLP
Store my watch	No	Yes
Where did I keep my watch?	No	Yes
Find my object	Yes	Yes
Add my object	Yes	Yes
Locate my watch	No	Yes
Total Utterances Accepted	2	5
Accuracy	20%	100%

Thus it shows that Natural Language Processing and Video Processing improves the efficiency of the system.

VI. CONCLUSION AND FUTURE SCOPE

In this paper, we develop an alexa skill that enables users to track items in their household. Thus, initially, we began by incorporating speech recognition algorithm in order to understand the context of user and perform appropriate tasks. In this study, we also used video processing algorithm that helps identify objects with the use of cameras around the house. The data obtained from users are stored and retrieved using Dynamo DB - A database provided by AWS. We also incorporated dialog management so as to improve user experience. This developed skill will be beneficial for older people who often forget the location of their belongings.

However, this skill has one limitation that there is no way for the skill to understand who is asking which item. In order to overcome this limitation, in the future, voice recognition will be implemented so as to increase the security of the skill. With voice recognition, the skill will be able to predict which user is accessing the skill based on their voice and provide personalized answers. As a result, security will also increase.

ACKNOWLEDGMENT

First and foremost, we wish to express our deep unfathomable feeling, gratitude to our institution and our department for providing us a chance to fulfill our long cherished dreams of becoming Computer Science

Engineers.We express our sincere thanks to our honorable Secretary Dr.C. Ramaswamy for providing us with required amenities. We wish to express our hearty thanks to Dr.A. Rathinavelu, Principal of our college, for his constant motivation and continual encouragement regarding our project work. We are grateful to our guide Dr.G.Anupriya, Professor and Head of the Department, Computer Science and Engineering, for her constant support and guidance offered to us during the course of our project.

REFERENCES

[1] Hyunji Chung, Sangjin Lee , *"Intelligent Virtual Assistant knows your Life",* Computers and Society, Vol. 42, pp. 201–213, 2017.
[2] Abhay Dekate, Chaitanya Kulkarni , Rohan Killedar , *"Study of Voice Controlled Personal Assistant Device"*, International Journal of Computer Trends and Technology(IJCTT),Vol.42, Issue.1, pp. 52–63, 2016.
[3] Veton Kepuska , *"Comparing Speech Recognition Systems(Microsoft API,Google API and CMU Sphinx)"*, International Journal of Engineering Research and Application, Vol.07, Issue.3(Part-2), pp. 20-24, 2017.
[4] Ashiq Anjum, *"Video Stream Analysis in Clouds: An Object Detection and Classification Framework for High Performance Video Analytics",* Proceeding of Transactions on Cloud Computing, University of Derby, United Kingdom, pp. 125–218.

Authors Profile

Ms. Sujitha Perumal pursed Bachelor of Engineering from Dr. Mahalingam College of Engineering and Technology(Autonomous), Pollachi under Anna University in 2019.

Mr. Mohammed Saqib Javid pursed Bachelor of Engineering from Dr. Mahalingam College of Engineering and Technology (Autonomous), Pollachi under Anna University in 2019. He is currently working in a Software firm as the Application Developer.

