1. Introduction

1.1 Objective

The focus of this project is to develop a robust question answering system in the agriculture domain. It aims to help farmers get information and resolve queries related to growing crops and thereby improving agriculture literacy. The system is based on the principles of speech recognition, natural language processing and information retrieval. Most of the currently available information retrieval tools return a list of ranked documents (indirect responses that are time consuming to check) instead of precise answers and do not support runtime answer retrieval. The Artificial Agriculture Officer, on the other hand, is a system which processes unstructured data and returns actual answers for questions such as `which', `what', `who', `where'. For example, "What kinds of diseases affect rice?"

Farmers may also face many problems related to plant disease and infection, and he may not always recognize the disease or know how to eradicate it. In this case, the application has an image recognition component that will be able to recognize various plant problems and will advise the farmer on how best to deal with it.

1.2 Problem Definition

The main motivation of this project is to facilitate the lives of the farmers. In the past, the most common and widespread way of gathering information was to approach a real-life Agriculture Officer. This had many issues: (a) The officer may not be all that knowledgeable (b) The officer may not always be available (c) The farmer may have to travel very long distances to meet the officer (d) The farmer may not be able to ask all his queries in one meeting. This caused a terrible inconvenience to the farmer, and instead of approaching the officer, he may try to farm using his own partial knowledge. This could lead to a lower crop yield, which is not desired.

Ever since the internet came into widespread use, it offered another way to get the required information. Farmers could rely on powerful search engines at home instead of having to physically travel to meet a person. However, this solution had its own problems. The internet usually responds to any question with long and winding answers (spread out across multiple webpages) that the farmer has to search through to find the relevant information that he set out to look for. Also, considering many farmers are illiterate, this poses a huge problem.

Currently, there are a few apps and phone services such as AgriSync and KisanSuvidha that are trying to rectify the above stated problems. However, they may not be the most optimal solutions. The Artificial Agriculture Officer is an attempt to effectively solve the above stated problems by providing short and succinct responses to any queries being asked and by providing an image recognition component that recognizes crop diseases.

1.3 Existing System

1.3.1AgriSync:

AgriSync enables farmers and advisors to connect and resolve support issues on the farm. Farmers can connect with multiple advisors from different companies to submit and receive support in real-time via video. Advisors can manage multiple service tickets through a dashboard and remote video that allows them to see what the farmer sees in real-time. A Web-based customer service dashboard allows the advisor's organization to see open cases, resolution status and farmer feedback in real-time.

How it works:

1. Advisors sign up and download the app in five minutes or less.
2. Invite company team members to join AgriSync from the mobile app or web dashboard.
3. Invite your farmer customers to use AgriSync before they need to request your help.
4. A farmer requests help from his local, trusted advisor by submitting a ticket with an issue description attaching any text, images and videos.
5. The advisor receives a text alert about an issue that needs resolved
6. When the advisor is ready to connect via live chat, the advisor has control of the call and will start the live conversation. Advisors see exactly what the farmer is seeing by controlling the camera. Capture images within the live call that can be automatically added to ticket details for a later reference.
7. After the conversation is complete, you will end the call and provide feedback about the quality of the call. Farmers provide feedback about the quality of service received from advisors and advisors have the ability to mark the call as “flagged” for further attention or “billable” and will receive instant feedback about your customer service interaction.
8. Check out the web dashboard at any time to download all customer service data that has been tracked.

1.3.2 KisanSuvidha-

Farmers often struggle for basic information like weather updates, crop prices and expert advice, ending up often relying on hearsays. The mobile app, KisanSuvidha, launched by Prime Minister NarendraModi will prove helpful for farmers to get the data, but they must own a smartphone.

The app has a simple interface and provides information on five critical parameters—weather, input dealers, market price, plant protection and expert advisories. An additional tab directly connects the farmer with the Kisan call centre where technical graduates answer their queries. The design is simple and neat.

To begin with, a farmer has to register the mobile number, choose a language, at present limited to Hindi and English, and enter details of the state, district and block or sub-district. A tap on the weather button shows details of temperature, humidity, wind and rainfall for the current day and the forecast for the next five days. Additionally, a farmer can get extreme weather alerts like hailstorms or unseasonal rains. For instance, after harvesting, farmers often leave their cereal crops in the field to dry. Prior information on freak rains can help them save their crop. The market price button shows latest price of all crops traded in a *mandi* or registered agriculture market of the particular district a farmer belongs to. Additionally, he gets to see the maximum price in the district, state and the entire country on a particular day.

Other information points are useful too. The plant protection button gives pest, weed and disease-related information as well as management practices for each stage of crop development—from nursery to harvesting. A farmer from, say Ganganagar district in Rajasthan, can scan the information for all major crops grown in the district like mustard, wheat and vegetables.

The agro advisory section shows messages for farmers from district agriculture officials and state universities in their local language. These primarily deal with crop management practices based on the prevailing situation, say a remedy for a widespread pest attack or imminent showers.
Farmers can also access names and mobile numbers of input dealers selling pesticides, seeds, fertiliser and machinery. This is a handy tool—farmers can now make a call and compare prices and availability before they actually head out to purchase these inputs.

1.4 Proposed System

The application being built will have a speech recognition component to get the input question from the user. It will then use Natural Language Processing (NLP) and DialogFlow to understand the question (conversion of a naturally phrased question into a database query) being asked and retrieve the appropriate answer from the knowledge-base. It also comes with an image recognition component to facilitate recognition of various plant based diseases.

Speech recognition is the inter-disciplinary sub-field of computational linguistics that develops methodologies and technologies that enables the recognition of spoken language into text by computers Recognizing user questions in natural languages involves Natural Language Processing (NLP).

NLP plays a big role in Information and Communications Technology (ICT) and Question Answering (QA) systems. Natural language processing (NLP) is the automated approach to scrutinizing text based on both a set of technologies and a set of theories. Rather than the keyword based retrieval methods, it has become significant to be able to ask queries and get answers, using natural language (NL) expressions. The QA system can better fulfil the needs of users as they provide a faster and more successful way of giving answers to user questions.

Image recognition and image classification is an easy task for anyone with a functioning brain. However, for a computer, it is a very difficult problem. As a human grows up, the brain subconsciously starts collecting visual information and making a large data store of objects and their relationships with the environment. If a child constantly comes across a variety of cars, eventually he will be able to distinguish between the different types, brands, and models. It would be naturally assumed, that if a child is able to learn to classify objects quickly, that it should be quite easy for a machine to as well. However, computers have a hard time differentiating images since it has to consciously take all factors into consideration such as lighting, size, color, etc.

1.5 Organization of Report

This thesis is organized as follows:

- Chapter I deals with the Introduction to the project, objective, motivation, problem statement and contributions.
- Chapter II explains the Literature Survey conducted before starting the project.
- Chapter III gives the design details of the system, like the overall architecture and design diagrams (UML).
- Chapter IV explains the implementation of the system. It discusses in detail about how the system is actually built.
- Chapter V deals with the result analysis.
- Chapter VI discusses conclusions and the future scope of the project
- References followed by the Appendix consisting of sample code segments

2. Literature Survey

2.1 Speech Recognition

Speech is the most basic, common and efficient form of communication method for people to interact with each other. People are comfortable with speech therefore it is reasonable to assume that they would also like to interact with computers via speech, rather than using primitive interfaces such as keyboards and pointing devices.

Classification of Speech Recognition Systems:

Speech recognition systems can be separated in several different classes by describing the type of speech utterance, type of speaker model, type of channel and the type of vocabulary that they have the ability to recognize. Speech recognition is becoming more complex and a challenging task because of this variability in the signal.

A. Types of Speech Utterance

An utterance is the vocalization (speaking) of a word or words that represent a single meaning to the computer. Utterances can be a single word, a few words, a sentence, or even multiple sentences. The types of speech utterance are:

1) Isolated Words Isolated word recognizers usually require each utterance to have quiet on both sides of the sample window. It doesn't mean that it accepts single words, but does require a single utterance at a time. This is fine for situations where the user is required to give only one word responses or commands, but is very unnatural for multiple word inputs. It is comparatively simple and easiest to implement because word boundaries are obvious and the words tend to be clearly pronounced which is the major advantage of this type. The disadvantage of this type is choosing different boundaries affects the results.

2) Connected Words Connected word systems (or more correctly 'connected utterances') are similar to isolated words, but allow separate utterances to be 'run-together' with a minimal pause between them.

3) Continuous Speech Continuous speech recognizers allow users to speak almost naturally, while the computer determines the content. Basically, it's computer dictation. It includes a great deal of "co articulation", where adjacent words run together without pauses or any other apparent division between words. Continuous speech recognition systems are most difficult to create because they must utilize special methods to determine utterance boundaries. As vocabulary grows larger, confusability between different word sequences grows.

4) Spontaneous Speech This type of speech is natural and not rehearsed. An ASR system with spontaneous speech should be able to handle a variety of natural speech features such as words being run together and even slight stutters. Spontaneous (unrehearsed) speech may include mispronunciations, false-starts, and nonwords.

B. Types of Speaker Model

All speakers have their special voices, due to their unique physical body and personality. Speech recognition system is broadly classified into two main categories based on speaker models namely speaker dependent and speaker independent.

1) Speaker dependent models Speaker dependent systems are designed for a specific speaker. They are generally more accurate for the particular speaker, but much less accurate for other speakers. These systems are usually easier to develop, cheaper and more accurate, but not as flexible as speaker adaptive or speaker independent systems.

2) Speaker independent models Speaker independent systems are designed for variety of speakers. It recognizes the speech patterns of a large group of people. This system is most difficult to develop, most expensive and offers less accuracy than speaker dependent systems. However, they are more flexible.

C. Types of Vocabulary

The size of vocabulary of a speech recognition system affects the complexity, processing requirements and the accuracy of the system. Some applications only require a few words (e.g. numbers only), others require very large dictionaries (e.g. dictation machines). In ASR systems the types of vocabularies can be classified as follows.

- Small vocabulary - tens of words
- Medium vocabulary - hundreds of words
- Large vocabulary - thousands of words
- Very-large vocabulary - tens of thousands of words
- Out-of-Vocabulary- Mapping a word from the vocabulary into the unknown word

Apart from the above characteristics, the environment variability, channel variability, speaking style, sex, age, speed of speech also makes the ASR system more complex. But the efficient ASR systems must cope with the variability in the signal.

2.2 Image Recognition

Image recognition generally has six main steps:

1. Image Formatting: This step consists of actually acquiring the image and converting it into a digital format
2. Conditioning: There are many parts of an image that are unnecessary for image recognition that were either added to the picture (as noise) during the process of digitization or they form part of the background. The conditioning step supresses or normalizes these uninteresting elements and highlights the more interesting parts.
3. Labelling: Images generally have informative patterns that can be used to identify various objects. Patterns consist of adjacent pixels that share certain characteristics such that it can be deduced that they belong to the same structure. Edge detection techniques use the assumption that continuous adjacent pixels that have a great difference in intensity or color mark the boundary between objects or objects and the background. Since, all discovered edges may not be important, the process of Thresholding filters out most of the insignificant edges. The edges considered important are then labelled.
4. Grouping: Grouping turns the primitive edges found in the labelling step into lines by determining which of the edges belong to the same spatial event. In the first few steps the image is considered as a digital picture (consisting of pixel information). However from grouping, the information regarding which spatial event that each pixel belongs to is stored in a logical data structure.
5. Extracting: Extracting refers to feature extracting which includes creating a list of properties the each of the groups of pixels in the same spatial event. This list can consist of area, grey tone moments, inscribing circle, a group's centroid, etc. The properties of each group depend on whether it is considered an arc or a region. For a region, it is more useful to make note of the number of holes. In the case of an arc, its curvature would be important to know. Feature extraction also takes topological relationships among the objects into consideration. Do they touch? Where are they in relation to one another? Etc.
6. Matching: The final step is to actually recognize the elements in the image. This is done by comparing the objects to a prior store of models and finding the best match (Template matching).

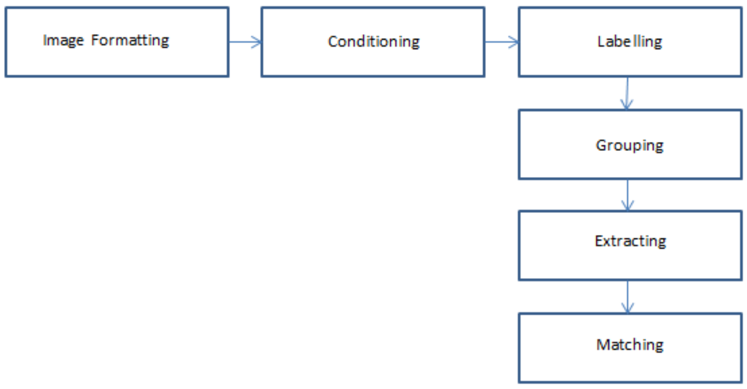


Figure 2.1: Image Recognition Flowchart

Image Recognition Approaches:

1) On The Basis Of Characteristic Used:
   - A. Shape-based: These methods make use of the objects' 2D spatial information. Common features used in shape-based classification schemes are the points (centroid, set of points), primitive geometric shapes (rectangle or ellipse), skeleton, silhouette and contour.
   - B. Motion-based: These methods use temporal tracked features of objects for the classification.

2) On The Basis Of Training Sample Used:
   - A. Supervised Classification: The process of using samples of known informational classes (training sets) to classify pixels of unknown identity. Example: minimum distance to means algorithm, parallelepiped algorithm, maximum likelihood algorithm
   - B. Unsupervised Classification: In this type of classification is a method which examines a large number of unknown pixels and divides it into number of classes based on natural groupings present in the image values. Computer determines spectrally separable class and then defines their information value. No extensive prior knowledge is required. Example: Kmeans clustering algorithm.

3) On The Basis Of Assumption Of Parameter on Data:
   - A. Parametric classifier: The parameters like mean vector and covariance matrix are used. There is an assumption of Gaussian distribution. The parameters like mean vector and covariance matrix are frequently generated from training samples. Example: Maximum likelihood, linear discriminant analysis.

B. Non Parametric classifier: There is no assumption about the data. Non-parametric classifiers do not make use of statistical parameters to calculate class separation. Example: Artificial neural network, support vector machine, decision tree classifier, expert system.

4) On The Basis Of Pixel Information Used:

A. Per pixel classifier: Conventional classifier generates a signature by using the combination of the spectra of all training-set pixels from a given feature. the contributions of all materials present in the training-set pixels is present in the resulting signature. It can be parametric or nonparametric the accuracy may not meet up because of the impact of the mixed pixel problem. Example: maximum likelihood, ANN, support vector machine and minimum distance.

B. Subpixel classifiers: The spectral value of each pixel is assumed to be a linear or non-linear combination of defined pure materials called end members, providing proportional membership of each pixel to each end member. Subpixel classifier has the capability to handle the mixed pixel problem, suitable for medium and coarse spatial resolution images. Example: spectral mixture analysis, subpixel classifier, Fuzzy-set classifiers.

C. Per-field classifier: The per-field classifier is intended to handle the problem of environmental heterogeneity, and also improves the classification accuracy. Generally used by GIS-based classification approaches. D. Object-oriented classifiers: Pixels of the image are united into objects and then classification is performed on the basis of objects. It involves 2 stages: image segmentation and image classification Image segmentation unites pixels into objects, and a classification is then implemented on the basis of objects. Example: e Cognition.

5) On The Basis Of Number Of Outputs For Each Spatial Element:

A. Hard Classification: Also known as crisp classification. In this each pixel is required or forced to show membership to a single class.eg maximum likelihood, minimum distance, artificial neural network, decision tree, and support vector machine.

B. Soft classification: also known as fuzzy classification. In this each pixel may exhibit numerous and partial class membership. Produces more accurate result.

6) On The Basis Of Spatial Information

1. Spectral Classifiers: This image classification uses pure spectral information. Example: Maximum likelihood, minimum distance, artificial neural network.
2. Contextual Classifiers This image classification uses the spatially neighbouring pixel information. Example: frequency-based contextual classifier.
3. Spectral-contextual classifiers: This classification uses both spectral and spatial information initial classification images are generated using parametric or non-parametric classifiers and then

contextual classifiers are implemented in the classified images. Example: combination of parametric or non-parametric and contextual algorithms.

7) Multiple classifiers approach:

Different classifiers have their own advantages and disadvantages. In this approach different classifiers are combined. some of the method for combining multiple classifier are: Voting rules, Bayesian formalism, evidential reasoning, multiple neural network.

Image Recognition Techniques:

Artificial Neural Network:

Artificial Neural Networks are computer systems that were designed based on the working of the human brain. It uses artificial neurons to model the tens of thousands of complex connections of biological neurons in the brain. In real life, a brain has billions of neurons each forming hundreds of paths as humans learn new things. In ANNs there are only a few layers of neurons that are much simpler each having few connections. Each connection (modelled after biological synapse) between the artificial neurons transmits signals from one to another. The recipient of the signal processes it and produces a response that is sent to the next neuron in the network. At the end layer an output is generated.

ANNs “learn” by way of examples. For example, in image recognition, they will look at hundreds of images of an object (without any prior knowledge about it) and learn to identify the same object in other images. The performance and accuracy of such systems depends on the structure of the network and the number of nodes.

Advantages:

- It is a non-parametric classifier
- It is a data driven self-adaptive technique
- It efficiently handles noisy inputs
- The computation rate is high

Disadvantages:

- It is semantically poor
- Training is time consuming
- Over-fitting
- It can be difficult to choose the network architecture

Decision Trees:

Decision trees are classification support tools that use a tree like data structure to help make decisions. They use various features of a given data (training data set) to learn about the most probable outcome for a test data set. Decision trees consist of a collection of nodes and branches in a tree like structure. Each internal node represents a variable with different representations of it. Each branch represents each of the representations. The leaf nodes represent class labels. In image recognition, the goal is to build a tree that accurately classifies the given images.

Advantages

- Handles nonparametric training data
- Does not require extensive design or training.
- Provides hierarchical associations
- Provides a set of rules that are easy to interpret
- Simple
- Computationally efficient

Disadvantages

- Tree structure is prone to sampling
- Splitting is locally greedy
- Instability
- Complexity
- Cost
- Prone to over-fitting

Support Vector Machines:

A support vector machine is a supervised learning model that analyzes data used for classification and regression analysis. Given a set of training examples, an SVM can build a model that exclusively categorizes different types of objects. It is considered to be a probabilistic binary linear classifier. Once the model has been thoroughly trained, new data sets can be given to the machine to be classified.

Support Vector Machines can also use unsupervised learning to build classification models. In unsupervised learning, the training data is absent. In other words, prior labels are not given to the objects. The machine "learns" by looking for similarities between the given data points and putting them in together in a way that makes the most sense.

A SVM separates the different categories of objects by creating a hyperplane or a set of hyperplanes in the n-dimensional space. A good separation is one that has the largest distance from each of the categories.

Advantages

- It offers flexibility over the choice in the form of the threshold
- Contains a nonlinear transformation
- It provides a good generalization capability
- The problem of over fitting is eliminated
- Reduction in computational complexity

Disadvantages

- Result transparency is low
- Training is time consuming
- Structure of algorithm is difficult to understand
- Determination of optimal parameters is not easy when there is nonlinearly separable training data.

Fuzzy Classification

Fuzzy classification is the categorization of objects into fuzzy sets based on a quantity called the truth value, which is defined by a membership function. The membership function is a function that indicates whether an object belongs to a certain set and by how much. If the result generated by the function is 1, the object completely belongs to the set. If the truth value generated is 0, it shows that the object does not have any characteristics that may associate it with a particular set. Any value in between shows a partial membership. A class is a set that is defined by a certain property, and all objects having that property are elements of that class. Classification is the process of grouping individuals having the same characteristics into a set. In fuzzy image recognition, stochastic associations are used to describe the qualities of a picture. These associations are used to determine which fuzzy set the image may belong to.

Advantages

- Can handle uncertainty
- Characteristics are described by discerning stochastic relationships.

Disadvantages

- Without prior knowledge the produced output is not efficient
- Precise solutions depend upon the direction of decisions

Tensorflow

Tensorflow is a tool for image recognition mainly designed using deep neural network models, specifically Convolutional Neural Networks. The latest model built by tensorflow implementing image recognition is the Inception V-3 model. It is trained to recognize various images from the ImageNet data set, which is an enormous image database, modelled after the WordNet, where each node represents an object defined by hundreds and thousands of images. Using Tensorflow as a tool has an advantage as it allows for transfer learning. Transfer learning refers to training an image recognition model that has already been trained on another data set or problem. This means that most of the processing that allows the model to distinguish objects has already been done. To make the model to recognize a new set of images, only the top-most layer must be re-trained. An alternative to the Inception V-3 model is the MobileNet, which gives more importance to optimization and being small than it does to accuracy.

Once the dataset has been created and imported into the Tensorflow directory, the next step is to create bottlenecks, which is the informal term used to refer to the layer of the network right before the last, which performs the actual classification. After all the bottlenecks have been created, the training of the final layer starts. The training works by supplying the cached value for each image into the bottleneck layer and the label for each image into a node called GroundTruth. These two inputs are necessary to calculate performance metrics of the training process

Table3.1 represents Performance Metrics of Model Training

Performance Metric	Definition
Training Accuracy	Percentage of images from current training batch labelled with correct class
Validation Accuracy	Precision of labelling on randomly selected images from another set
Cross Entropy	A function that indicates the progress of learning

To know the real measure of how well the model is performing is to see how well it distinguishes images not in its training data set. This is the validation accuracy. If the validation accuracy stays low while the training accuracy increases, this means that the network is over-fitting. The main purpose of the training is to make the cross entropy low.

The Tensorflow training process consists of 4000 training steps, where each step takes 10 random images from the training data set, finds their bottlenecks and sends them to the final layer to get the model's predictions. These predictions are then checked for validity by comparing them to the genuine labels. The weights in the network are updated accordingly. As the process continues, the accuracy increases. At the end of the script, a final test of accuracy is run using a small sample of the training data set that was kept separately. This evaluation is a marker of how well the trained model will perform the required classification.

Convolutional Neural Networks: A Convolutional Neural Network (CNN) is essentially a network that uses multiple copies of the same neuron. This allows for computationally large networks with a fairly low number of parameters (which are the values describing the behaviour of the neurons). This is similar to calling the same function many times in computer programming. CNNs require much less pre-processing than other image classification algorithms, which means that the network learns to process its own data before classification whereas other traditional algorithms need to be hard-engineered. A CNN consists of many hidden layers that can be made up of convolutional layers, pooling layers, fully connected layers, and normalization layers.

Convolutional Layer

The convolution layer performs a convolution operation on the input, and passes the result on the next layer. The convolution operation is a function that resembles the response of a biological neuron when it receives visual stimuli. Each of these convolutional neurons only processes data in its own receptive field. This type of operation offers a huge advantage over a fully connected feed-forward neural network in image classification, as an it requires fewer neurons and parameters.

Pooling Layer

Global or Local Pooling refers to the combination of outputs of one layer into a single neuron, being used as the input to the next.

- Max Pooling uses the highest value from each of a cluster of neurons.
- Average Pooling uses the average value of each cluster of neurons.

Fully Connected Layer

Fully connected layers are those where every neuron in one layer is connected to every neuron the next.

3. Methodology

3.1 System Design

3.1.1 Proposed Algorithm

- The user provides his authentication details (in this case phone number)
- The application sends an OTP to the user
- Upon verification of user, home page opens up
- If user gives speech input
  - System recognizes words
  - Understands query using natural language processing (dialogflow)
  - Searches knowledge-base for appropriate response
  - Returns verbal response
- Else if user gives visual input
  - System processes image
  - Searches database
  - System tries to match the image with previously “learnt” images
  - Returns a response on how to deal with particular disease
- User logs out

3.1.2 Diagrammatic representations

The Unified Modelling Language (UML) is a standard language for writing software blue prints. The UML is a language which provides vocabulary and the rules for combining words in that vocabulary for the purpose of communication. A modelling language is a language whose vocabulary and the rules focus on the conceptual and physical representation of a system. Modelling yields an understanding of a system.

UML Concepts

The Unified Modelling Language (UML) is a standard language for writing software blue prints. The UML is a language for:

- Visualizing
- Specifying
- Constructing
- Documenting the artifacts of a software intensive system.

The UML is a language which provides vocabulary and the rules for combining words in that vocabulary for the purpose of communication. A modelling language is a language whose vocabulary and the rules focus on the conceptual and physical representation of a system. Modelling yields an understanding of a system.

Building Blocks of the UML

The vocabulary of the UML encompasses three kinds of building blocks:

Things

Things are the abstractions that are first-class citizens in a model; relationships tie these things together; diagrams group interesting collections of things.
There are four kinds of things in the UML:

- Structural things
- Behavioral things
- Grouping things
- Annotational things

Relationships

There are four kinds of relationships in the UML:
A **dependency** is a semantic relationship between two things in which a change to one thing may affect the semantics of the other thing (the dependent thing).

An **association** is a structural relationship that describes a set links, a link being a connection among objects. Aggregation is a special kind of association, representing a structural relationship between a whole and its parts.

A **generalization** is a specialization/ generalization relationship in which objects of the specialized element (the child) are substitutable for objects of the generalized element (the parent).

A **realization** is a semantic relationship between classifiers, where in one classifier specifies a contract that another classifier guarantees to carry out.

Diagrams

UML has many types of diagrams, which are divided into two categories. Some types represent structural information, and the rest represent general types of behavior, including a few that represent different aspects of interactions. These diagrams can be categorized hierarchically as shown in the figure 3.1.

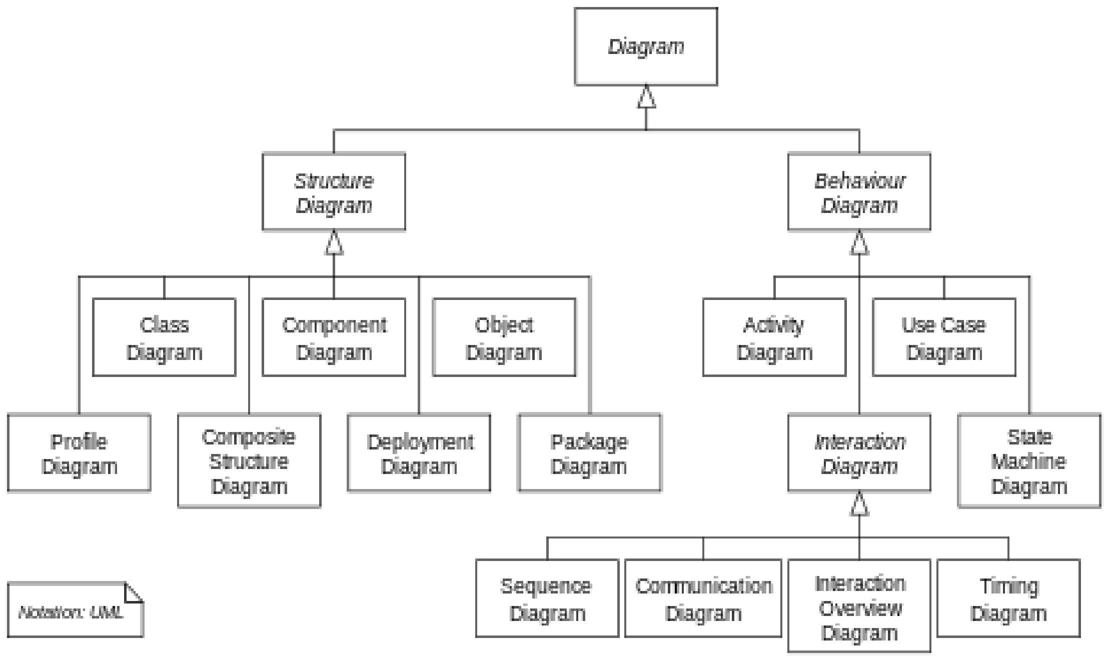


Figure 3.1: UML Diagrams

Structure diagrams emphasize the things that must be present in the system being modelled. Since structure diagrams represent the structure; they are used extensively in documenting the software architecture of software systems. For example, the component diagram describes how a software system is split up into components and shows the dependencies among these components. Behavior diagrams emphasize what must happen in the system being modelled. Since behavior diagrams

illustrate the behavior of a system, they are used extensively to describe the functionality of software systems. As an example, the activity diagram describes the business and operational step-by-step activities of the components in a system. Interaction diagrams, a subset of behavior diagrams, emphasize the flow of control and data among the things in the system being modelled. For example, the sequence diagram shows how objects communicate with each other in terms of a sequence of messages.

3.1.2.1 Data Flow Diagrams

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

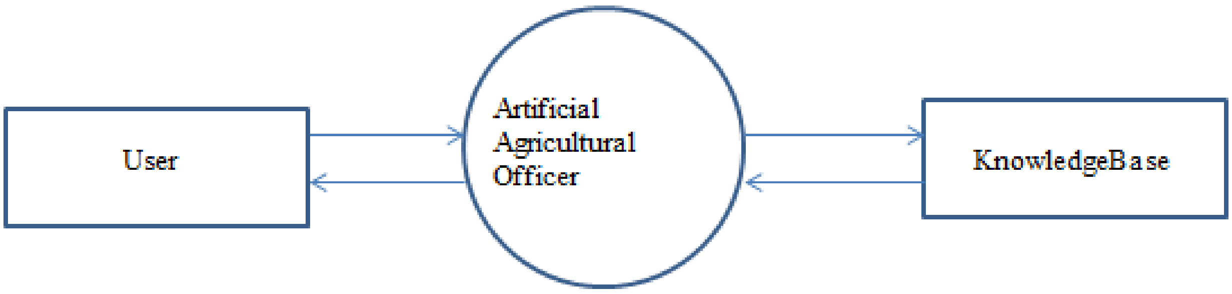


Figure 3.2: Data Flow Diagram Level 0 for Artificial Agriculture Officer

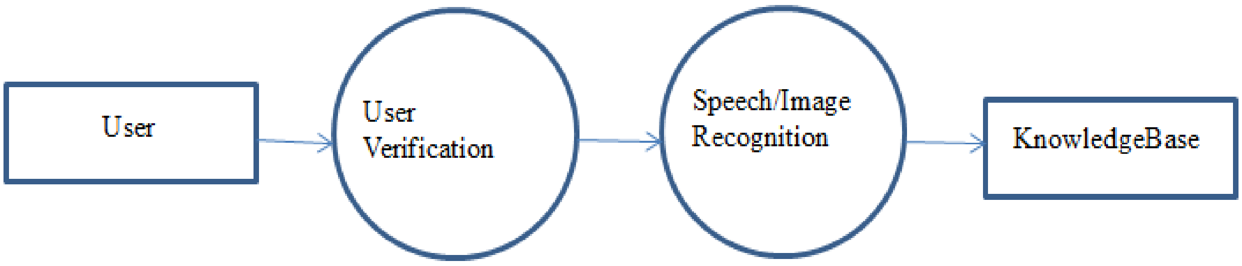


Figure 3.3: Data Flow Diagram Level 1 for Artificial Agriculture Officer

Data flow diagram for Artificial Agriculture Officer has 2 levels, Level 0 and Level 1.

- In Level 0, the user (Farmer) asks his question verbally. This query is transformed into a understandable query by the system and is then the answer is searched for through the database. The response is sent back to the user. The user receives a verbal answer.
- In Level 1, User (Farmer) asks his question verbally. This is recognized into words by speech recognition and is transformed into an understandable query by the system. The answer is then

searched for through the database. The response is sent back to the user. The user receives a verbal answer.

3.1.2.2 Use Case Diagram

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.

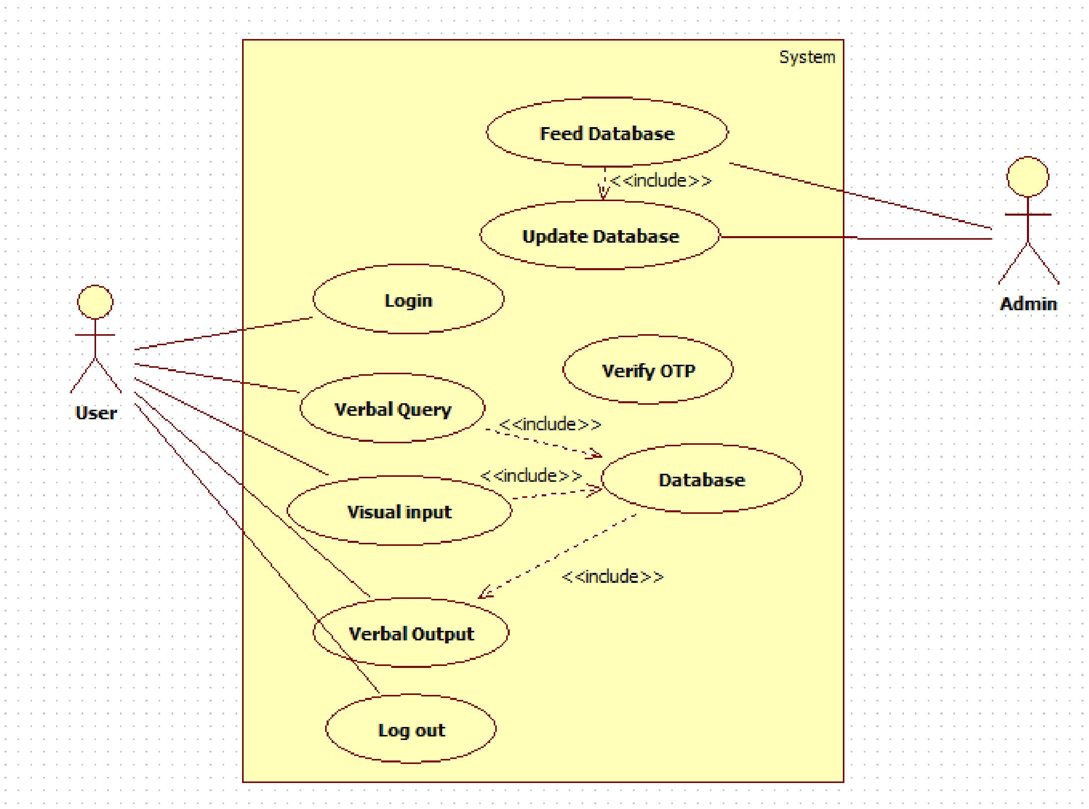


Figure 3.4: Use Case Diagram for Artificial Agriculture Officer

In the above diagram, the admin has tasks associated with managing the knowledgebase. He performs the use cases: “Feed Database” and “Update Database”. The User can “Login” which automatically verifies the user credentials and give either a verbal or visual input which gets searched for in the database. Finally, he can “Log out”.

3.1.2.3 Sequence Diagram

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams. Sequence diagrams demonstrate the behaviour of objects in a use case by describing the objects and the messages they pass. The diagrams are read left to right and descending.

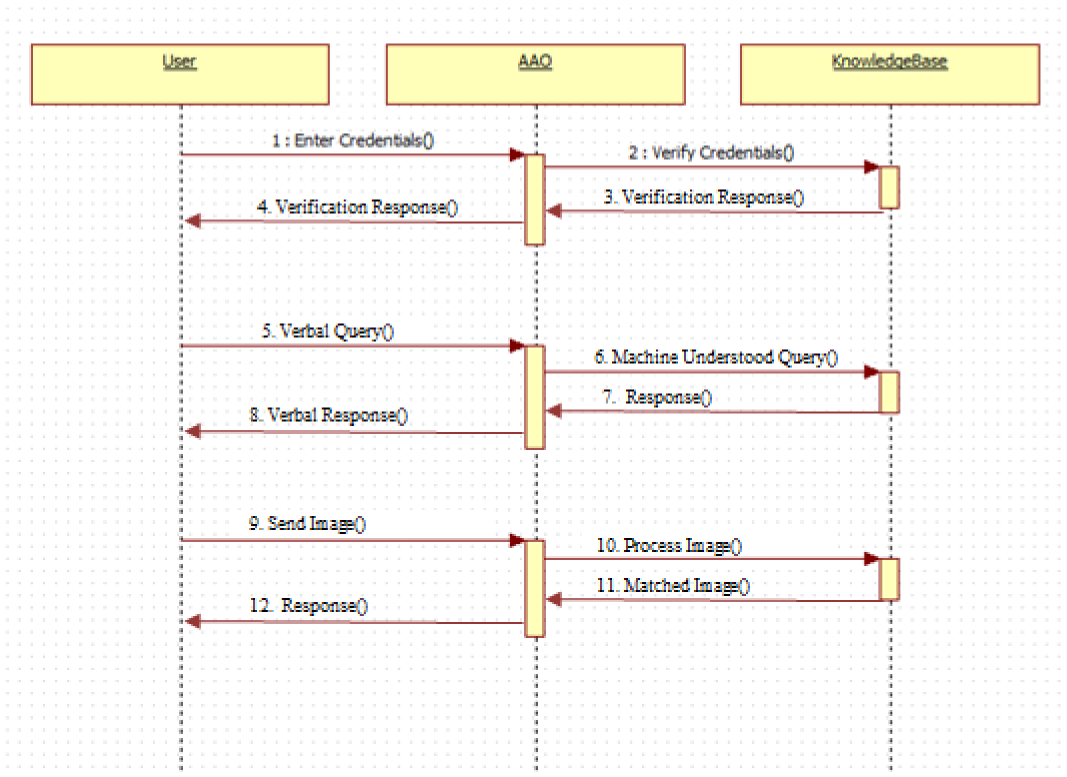


Figure 3.5: Sequence Diagram forArtificial Agriculture Officer

As shown in the diagram above, there are three main characters with a lifeline: the User, the Artificial Agriculture Officer (AAO), and the Knowledgebase. The user will send a message to the AAO to login. The AAO in turn will check with the knowledgebase for verification. If the user is verified, a response is sent back to the AAO and then the user. The user can then make a verbal query/visual query. The query is processed by the AAO, searched for in the knowledgebase, and a response makes its way back to the user.

3.1.2.4 Collaboration diagrams

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modelling

Language (UML). A collaboration diagram resembles a flowchart that portrays the roles, functionality and behavior of individual objects as well as the overall operation of the system in real time. Objects are shown as rectangles with naming labels inside. These labels are preceded by colons and may be underlined. The relationships between the objects are shown as lines connecting the rectangles. The messages between objects are shown as arrows connecting the relevant rectangles along with labels that define the message sequencing.

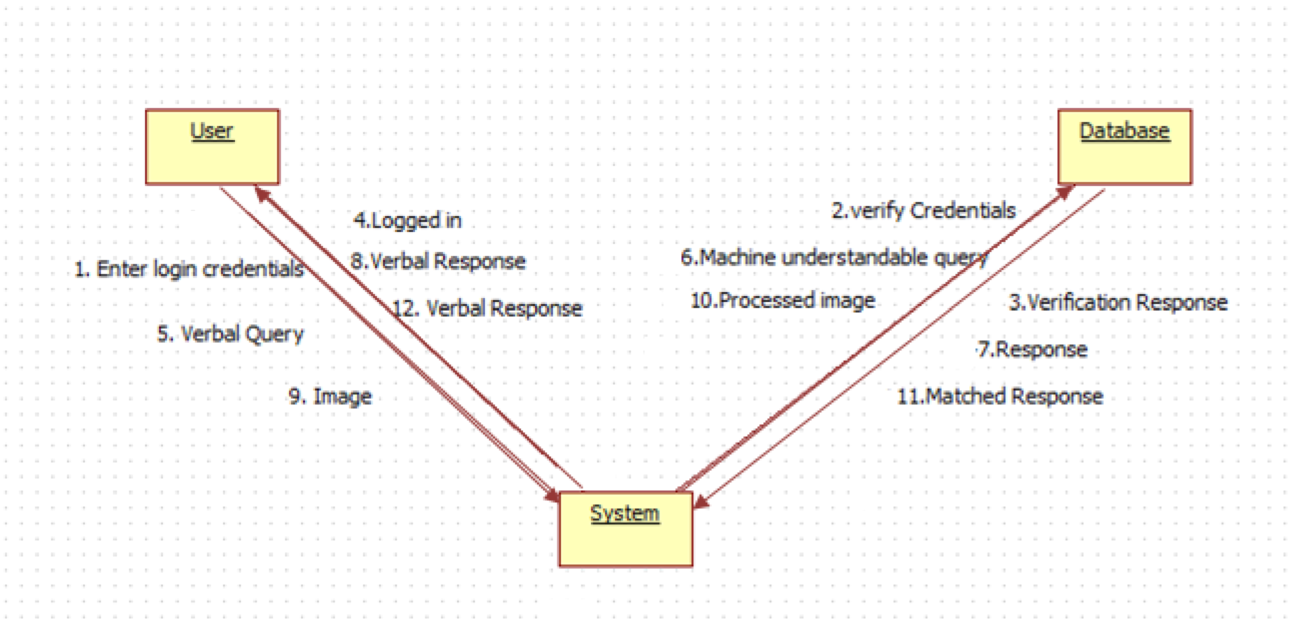


Figure 3.6: Collaboration Diagram for Artificial Agriculture Officer

As shown in the diagram above, there are three main characters: The User, the Artificial Agriculture Officer (AAO), and the Knowledgebase. The user will send a message to the AAO to login. The AAO in turn will check with the knowledgebase for verification. If the user is verified, a response is sent back to the AAO and then the user. The user can then make a verbal query/visual query. The query is processed by the AAO, searched for in the knowledgebase, and a response makes its way back to the user.

3.1.2.5 Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control.

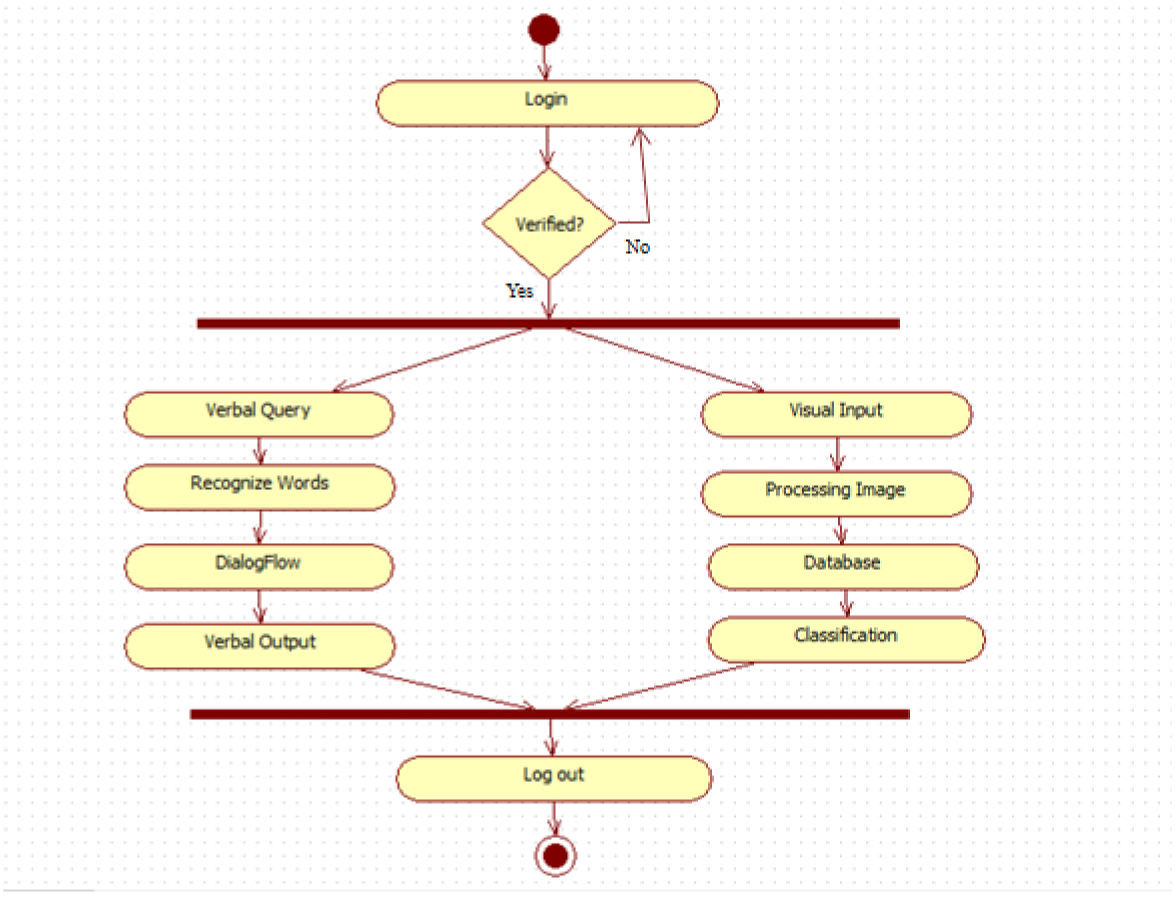


Figure 3.7: Activity Diagram for Artificial Agriculture Officer

In the above diagram, the initial activity is logging in. Upon logging in, the user credentials are verified. If the verification is successful, the user can either give the system a verbal or visual input. The words in the verbal input are recognized and then sent to the Dialogflow module, which then returns a verbal response. The image gets processed, and matched against the existing classification model. A classification response is returned. Finally, the user logs out.

3.1.2.6 State Machine Diagram

State chart diagrams are one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime. And these states are changed by events. In consequence, state chart diagrams are useful to model reactive systems.

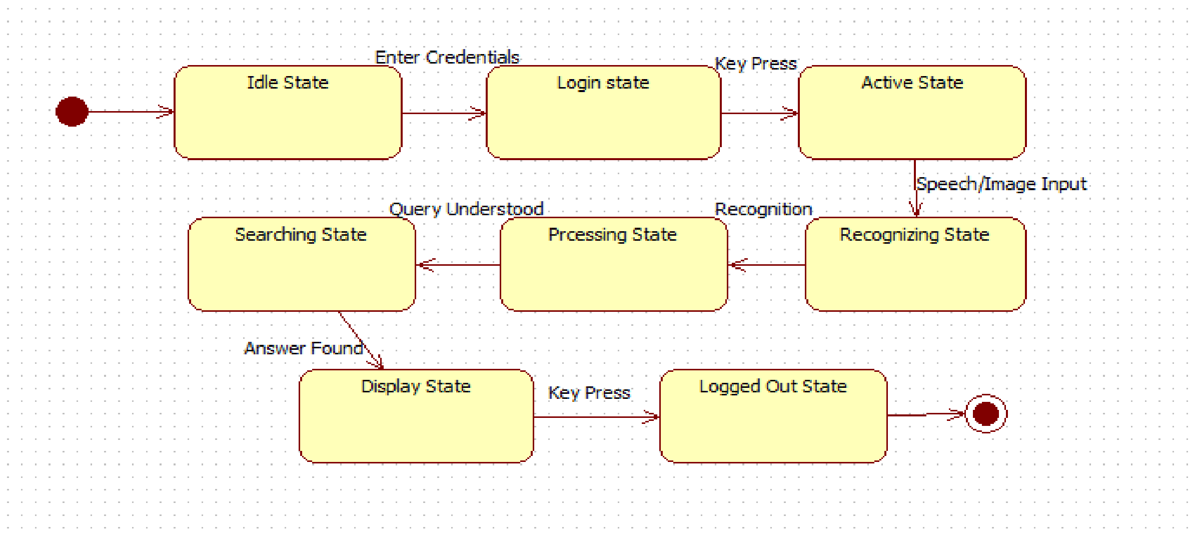


Figure 3.8: State Chart Diagram for Artificial Agriculture Officer

As shown in the diagram above, the system begins in an idle state. Upon entering valid credentials, it changes to the login state. Upon key press, it enters the active state, where it "actively" waits for either a verbal or visual input. Once it receives an input, it enters the recognizing state, where it tries to identify the characteristics of the input. Once the input is recognized, it enters the processing state, where necessary functions take place. Once the query and the type of response required is understood, the system enters the searching state, where it looks for the appropriate response. If an answer is found, it displays it. By pressing a key, the user can log out.

3.1.2.7 Component Diagram

A component diagram depicts how component are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems.

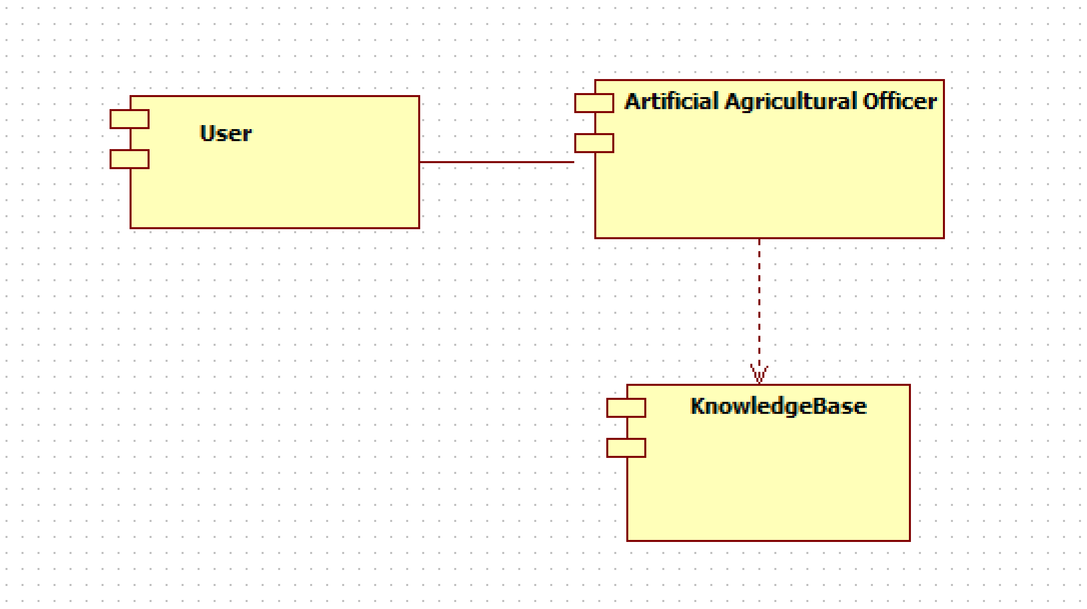


Figure 3.9: Component Diagram for Artificial Agriculture Officer

As shown in the diagram above, there are three main components: the User, the Artificial Agriculture Officer (AAO), and the Knowledgebase. The User is associated to the AAO, and the AAO is dependent on the data from the Knowledgebase.

3.1.2.8 Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed.

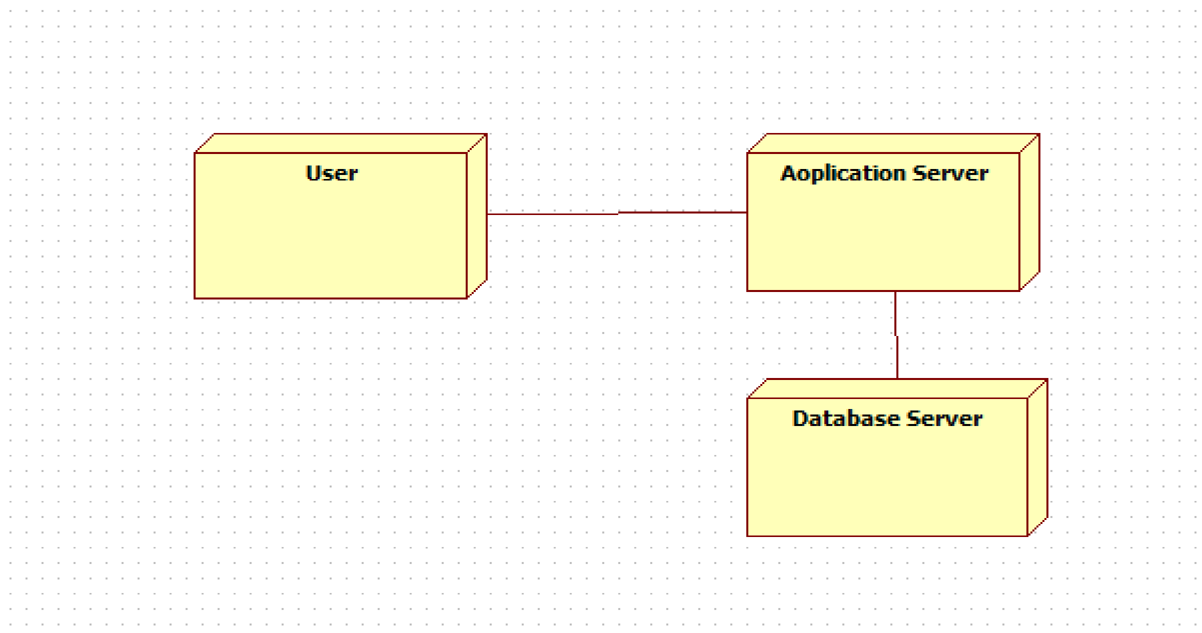


Figure 3.10: Deployment Diagram for Artificial Agriculture Officer

As shown in the diagram above, there are three main nodes: The User, the Application Server, and the Database Server. The User node is associated with the application server where the Artificial Agriculture Officer runs. The application server is associated with the database server where the database runs.

3.2 Implementation of Proposed solution

3.2.1 Module Description

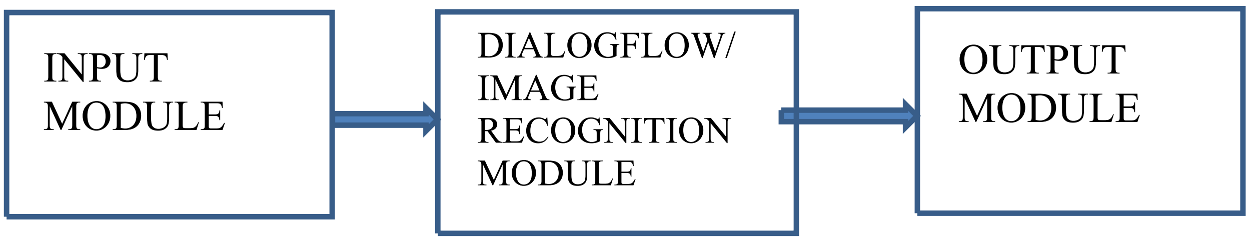


Figure 3.11: Module Flow Chart for Artificial Agriculture Officer

1. Input Module

- Speech:
  - Android comes with an inbuilt feature speech to text through which you can provide speech input to your app. As soon as a user say something, Android will recognise his/her voice and convert it into text. It will do it through Recognizer Intent. In the code Recognizer Intent is triggered which asks for speech input and then sends it through speech recognizer. It does it through ACTION_RECOGNIZE_SPEECH. If request code is REQ_CODE_SPEECH_INPUT, then corresponding text is written in output screen. In the background how voice input works is, the speech input will be streamed to a server, on the server voice will be converted to text and finally text will be sent back to our app.

- Image:
  - The Tensorflow module allows the user to provide a visual input to the application so that it can distinguish between various crop diseases.

2. DialogFlow/Image Recognition Module

DialogFlow

The Dialog Flow tool allows users to store data and create a flow of dialogue using JSON. Dialogflow works by allowing its users to create agents, which are NLU (Natural Language Understanding) modules that represent the overall purpose or topic of a conversation. Some prebuilt agents are available to work with: small talk, car, various language bots, etc.

Within each agent, the user can create multiple intents, which represents a mapping between what a user says and what action should be taken by your software. For each of the intents, the user must specify 'Training Phrases', 'Actions', 'Contexts', and 'Responses'.

Each agent can also have multiple entities. An entity is a powerful tool used for extracting parameter values from natural language inputs. Any important data you want to get from a user's request will have a corresponding entity. There are some prebuilt entities such as sys:location, sys:color, etc.

An action corresponds to the step your application will take when a specific intent has been triggered by a user's input. Actions can have parameters for extracting information from user requests and the response will appear in the JSON format.

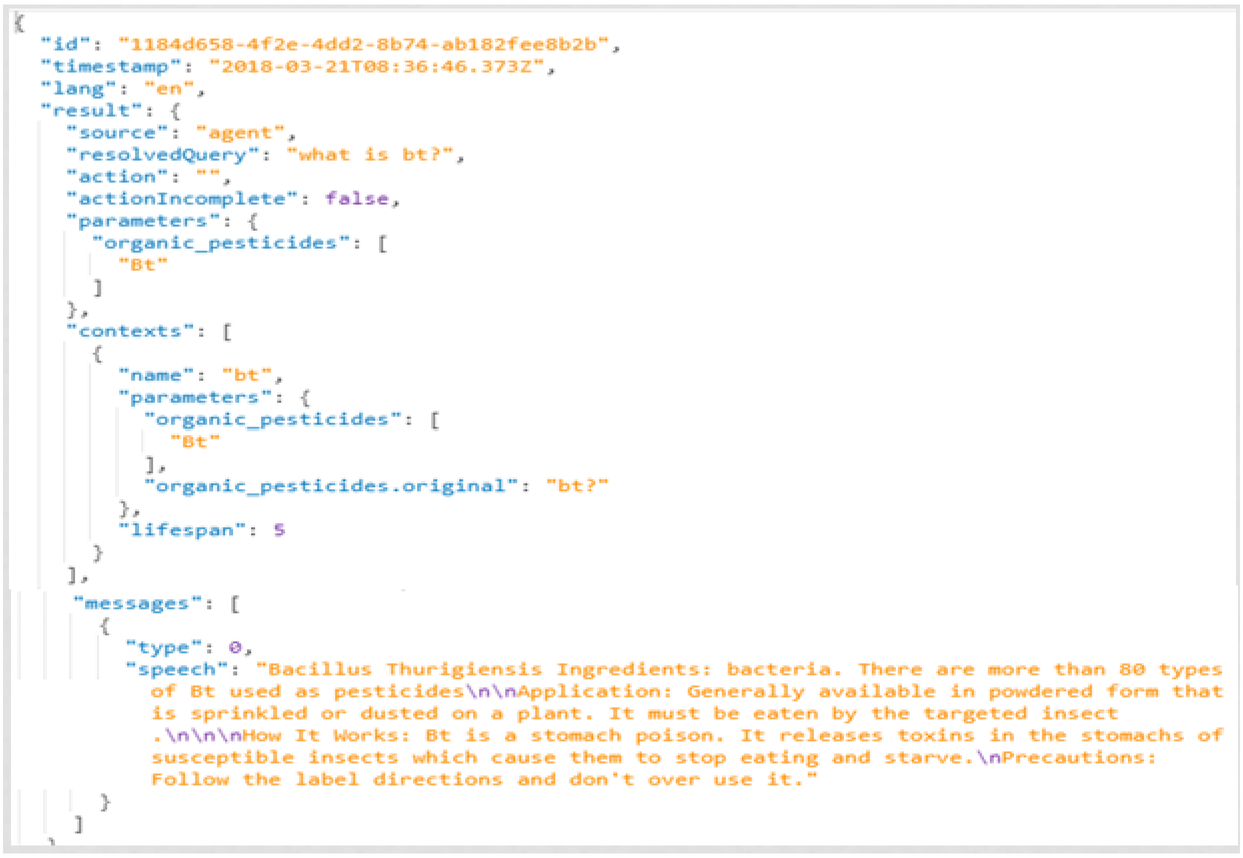
Contexts represent the current context of a user's request. This is helpful for differentiating phrases which may be vague or have different meanings depending on the user's preferences, geographic location, the current page in an app, or the topic of conversation

There are two types of dialogs to consider when building voice interaction scenarios:

Linear dialogs - the aim of which is to collect the information necessary to complete the required action (e.g. find the best hotel, turn on the right light bulb, or play the desired song)

Non-linear dialogs - which may have several branches, depending on users' answers

Since for each intent there can be multiple training phrases, a dataset is created in excel. The spreadsheet dataset must have the column names: IntentID, IntentName, Query (which represents the list of training phrases), and Response. A sample of the dataset created for the Artificial Agriculture Officer is shown in Figure 3.13.



```
{
  "id": "1184d658-4f2e-4dd2-8b74-ab182fee8b2b",
  "timestamp": "2018-03-21T08:36:46.373Z",
  "lang": "en",
  "result": {
    "source": "agent",
    "resolvedQuery": "what is bt?",
    "action": "",
    "actionIncomplete": false,
    "parameters": {
      "organic_pesticides": [
        "Bt"
      ]
    },
    "contexts": [
      {
        "name": "bt",
        "parameters": {
          "organic_pesticides": [
            "Bt"
          ],
          "organic_pesticides.original": "bt?"
        },
        "lifespan": 5
      }
    ],
    "messages": [
      {
        "type": 0,
        "speech": "Bacillus Thurigiensis Ingredients: bacteria. There are more than 80 types
          of Bt used as pesticides\n\nApplication: Generally available in powdered form that
          is sprinkled or dusted on a plant. It must be eaten by the targeted insect
          .\n\n\nHow It Works: Bt is a stomach poison. It releases toxins in the stomachs of
          susceptible insects which cause them to stop eating and starve.\nPrecautions:
          Follow the label directions and don't over use it."
      }
    ]
```

Figure 3.12: JSON object format in Dialogflow

The figure above shows the structure in which query data is stored within the Dialogflow. It is in JSON object format. Once a query has been made, a new object is generated. The first part of the object denotes the query ID, the time the query was made and the language it was made in. Next it stores the source of the query (in this case the agent), the query itself, and if any action was specified to be performed when it was called. If the query contains any parameters (in other words: the specified entities) it is stored in the 'parameters' variable. The next part of the object contains information about the context of the query (if it exists). The context concept allows the user to ask follow up questions about recognized parameters in the original query. For example, if the first question is: "What is Bt?", a follow up question can be: "What are the advantages of using it?". The system should automatically assume that the user is talking about "Bt" until the lifespan of the context expires. In the case shown above, the lifespan is 5 (which represents 5 follow up queries). The third part of the object contains the response. This is what the system responds with when the query is asked. The prefix "speech" tells the system that the response should be spoken aloud.

Tensorflow

Tensorflow is a tool for image recognition mainly designed using deep neural network models, specifically Convolutional Neural Networks. The latest model built by tensorflow implementing image recognition is the Inception V-3 model. It is trained to recognize various images from the ImageNet data set, which is an enormous image database, modelled after the WordNet, where each node represents an object defined by hundreds and thousands of images. Using Tensorflow as a tool has an advantage as it allows for transfer learning. Transfer learning refers to training an image recognition model that has already been trained on another data set or problem. This means that most of the processing that allows the model to distinguish objects has already been done. To make the model to recognize a new set of images, only the top-most layer must be re-trained. An alternative to the Inception V-3 model is the MobileNet, which gives more importance to optimization and being small than it does to accuracy.

3. Output Module

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech computer or speech synthesizer, and can be implemented in software or hardware products. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech.

Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output.

The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood clearly.

A text-to-speech system is composed of two parts:a front-endand a back-end.

The front-end has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called text normalization, pre-processing, or tokenization. The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end.

The back-end, often referred to as the synthesizer, then converts the symbolic linguistic representation into sound. In certain systems, this part includes the computation of the target prosody (pitch contour, phoneme durations), which is then imposed on the output speech.

3.2.2 Implementation Details

3.2.2.1 Text to Speech

After the preliminary steps of actually creating the android application and creating user interface elements, the next stage is to listen for user events. This is done by importing the following statements in the java file of the activity you want to implement text-to-speech in.

```
import android.view.View.OnClickListener;
import android.widget.Button;
import android.view.View;
```

The next step is to implement the onClickListener interface and call the onClick function. When the app receives the data from the backend, it will automatically call the onClick function and store the text. The app needs to check that the user has the data necessary for the TTS function before the user can call its methods. This requires the declaration and instantiation of the following instance variable at the top of the Activity class declaration, before the "onCreate" method:

```
Private int MY_DATA_CHECK_CODE = 0;
import android.content.Intent;
```

Finally, the app is ready to convert the input text to speech.

3.2.2.2 Authentication of Credentials using Firebase

Firebase is a tool that offers many different facilities, of which one is user authentication using phone numbers. Firebase Authentication can be used to sign in a user by sending an SMS message to the user's phone. The user signs in using a one-time code contained in the SMS message. Authentication using only a phone number, while convenient, is less secure than the other available methods, because possession of a phone number can be easily transferred between users. Also, on devices with multiple user profiles, any user that can receive SMS messages can sign in to an account using the device's phone number. To use phone number based sign-in in your app, it should offer it alongside more secure sign-in methods, and inform users of the security tradeoffs of using phone number sign-in.

The following are the steps for phone number authentication using firebase:

1. **Enable Phone Number sign-in for your Firebase project:**
   To use the phone number authentication method, first Phone Number sign-in in the firebase project has to be enabled.

2. **Send a verification code to the user's phone**:
   To initiate phone number, sign-in, present the user an interface that prompts them to type their phone number. Legal requirements vary, but as a best practice and to set expectations for your users, you should inform them that if they use phone sign-in, they might receive an SMS message for verification and standard rates apply. Then, pass their phone number to the PhoneAuthProvider. verifyPhoneNumber method to request that
   Firebase verify the user's phone number.
   For example:

```
PhoneAuthProvider.getInstance().verifyPhoneNumber(
  phoneNumber,              //phone number to verify
  60,                       //timeout duration
  TimeUnit.SECONDS,         //unit of timeout
  This,                     //activity(for callback binding)
  mCallbacks);              //OnVerficationStateChangedCallbacks
```

   This function has a number of variables that are required for it to work as shown in the comments. The verifyPhoneNumber method is reentrant: it can be called multiple times, but then it will not send a second SMS unless the original request has timed out.

   Once the number has been processed, there are two possible outcomes:

onVerificationCompleted(PhoneAuthCredential)

This method is called in two situations:

– Instant verification: in some cases, the phone number can be instantly verified without needing to send or enter a verification code.

– Auto-retrieval: on some devices, Google Play services can automatically detect the incoming verification SMS and perform verification without user action. (This capability might be unavailable with some carriers.) In either case, the user's phone number has been verified successfully, and you can use the PhoneAuthCredential object that's passed to the callback to sign in the user.

onVerificationFailed(FirebaseException)

This method is called in response to an invalid verification request, such as a request that specifies an invalid phone number or verification code.

3. **Create a PhoneAuthCredential object:**

   After the user enters the verification code that Firebase sent to the user's phone, create a PhoneAuthCredential object.

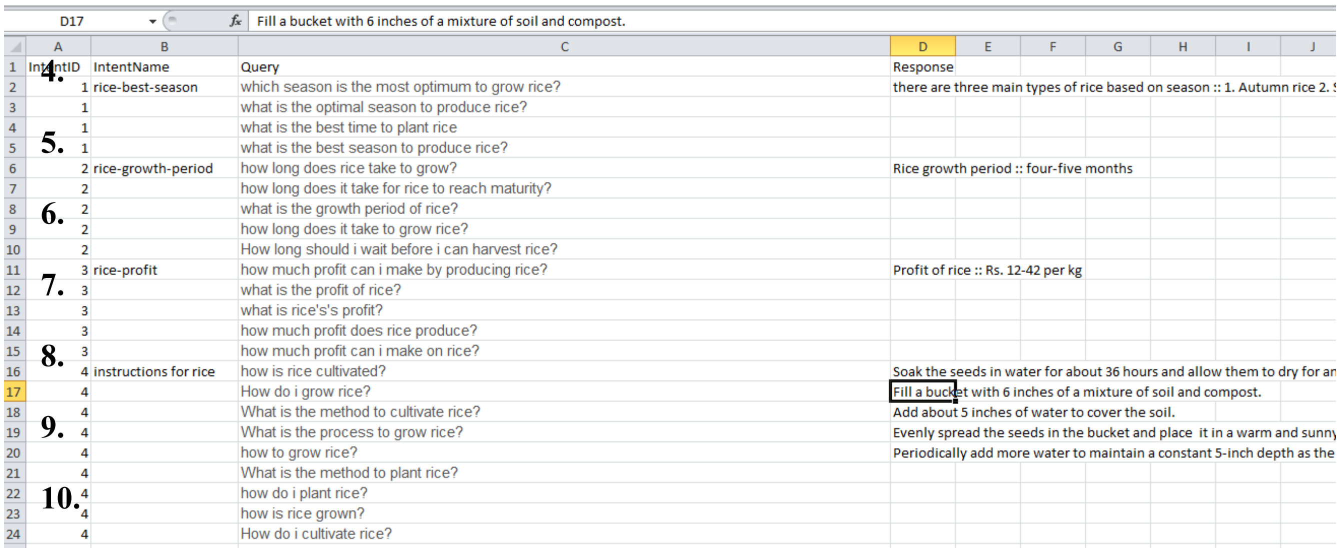
4. **Sign in the user:**

   After creating PhoneAuthCredential object complete the sign-in flow by passing the object PhoneAuthCredential to FirebaseAuth.signInWithCredential.

5. **Sign out the user:**

   To sign out a user, call signOut.
   FirebaseAuth.getInstance().signOut();

3.2.2.3 Adding data to Dialogflow



D17 Fill a bucket with 6 inches of a mixture of soil and compost.

	A	B	C	D
1	IntentID	IntentName	Query	Response
2	1	rice-best-season	which season is the most optimum to grow rice?	there are three main types of rice based on season :: 1. Autumn rice 2.
3	1		what is the optimal season to produce rice?	
4	1		what is the best time to plant rice	
5	1		what is the best season to produce rice?	
6	2	rice-growth-period	how long does rice take to grow?	Rice growth period :: four-five months
7	2		how long does it take for rice to reach maturity?	
8	2		what is the growth period of rice?	
9	2		how long does it take to grow rice?	
10	2		How long should i wait before i can harvest rice?	
11	3	rice-profit	how much profit can i make by producing rice?	Profit of rice :: Rs. 12-42 per kg
12	3		what is the profit of rice?	
13	3		what is rice's's profit?	
14	3		how much profit does rice produce?	
15	3		how much profit can i make on rice?	
16	4	instructions for rice	how is rice cultivated?	Soak the seeds in water for about 36 hours and allow them to dry for an
17	4		How do i grow rice?	Fill a bucket with 6 inches of a mixture of soil and compost.
18	4		What is the method to cultivate rice?	Add about 5 inches of water to cover the soil.
19	4		What is the process to grow rice?	Evenly spread the seeds in the bucket and place it in a warm and sunny
20	4		how to grow rice?	Periodically add more water to maintain a constant 5-inch depth as the
21	4		What is the method to plant rice?	
22	4		how do i plant rice?	
23	4		how is rice grown?	
24	4		How do i cultivate rice?	

Figure 3.13: Dialogflow Dataset

An IntentID must be specified once per row corresponding to the intent the row belongs to. The IntentName must be specified once per intent created. There can be any number of queries under the Query column (there can be any number of training phrases associated with a single intent. Increasing the number of training phrases helps the dialogflow to better respond to the user (it helps the dialogflow to associate the given question to an intent more efficiently). The number of responses can also vary per intent. When a particular query with multiple response options is made, any one of them can be displayed as the output.

The .csv file must first be converted to text. This can be done using an online converter. This text file is then uploaded to dialogflow under the sub-tab labelled, “Training”.

3.2.2.4 Retraining the mobilenet

Plant disease diagnosis system on Android is done by implementing a deep convolutional neural network with Tensorflow to detect disease from various plant leave images.Generally, due to the size limitation of the dataset, transefer learning is adopted in this system. Specifically, retrain the MobileNets, which is first trained on ImageNet dataset, on the plant disease datasets. Finally, we port the trained model to Android.

The first step in retraining the mobilenet is to create a directory.

A dataset is created, where hundreds of images of each category of object is stored in a separate folder. This folder has to copied into the tf_files file inside the newly created directory.

The architecture of the model being used must be specified to mobilenet.

Using a python command, "bottlenecks" are generated. Each bottleneck is a text file containing data that is required for classification.

Upon creation of the bottlenecks, two new files are added to the tf_files folder of the directory: retrained_graph.pb and retrained_labels.txt

- The retrained_graph.pb contains data regarding how the graph has been modified
- The retrained_labels.txt contains data regarding the classification labels of the objects.

Using a python command, the model can be tested to see how well it classifies "test" images.

If the training is successful, an android application can be created.

To optimize the training of the model, a python command is used to generate a file: optimized_graph.pb, this gives the model a 30% boost in speed in the classification.

Compressing the files created is optional.

The graph and label files must be copied into the assets folder in android which is located in the directory – this allows for the application to classify the customized dataset.

Once the modifications are made, the gradle is built and the application is run on an adroid device.

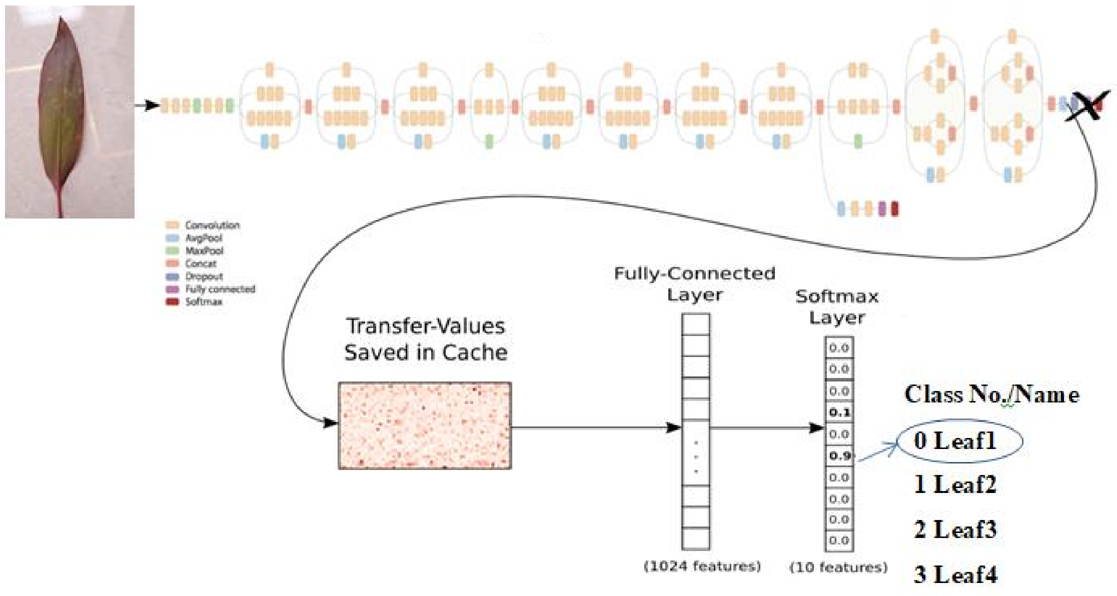


Figure 3.14: Training Model Using Tensorflow

3.3 System Requirements

Hardware and Software requirements:

1) Hardware:
   - Smartphone with inbuilt camera
   - Inbuilt microphone or external microphone
   - Headphones set (optional)
2) Software
   - The Artificial Agriculture Officer Application

4. Results and discussions

4.1 Output:

Login page:

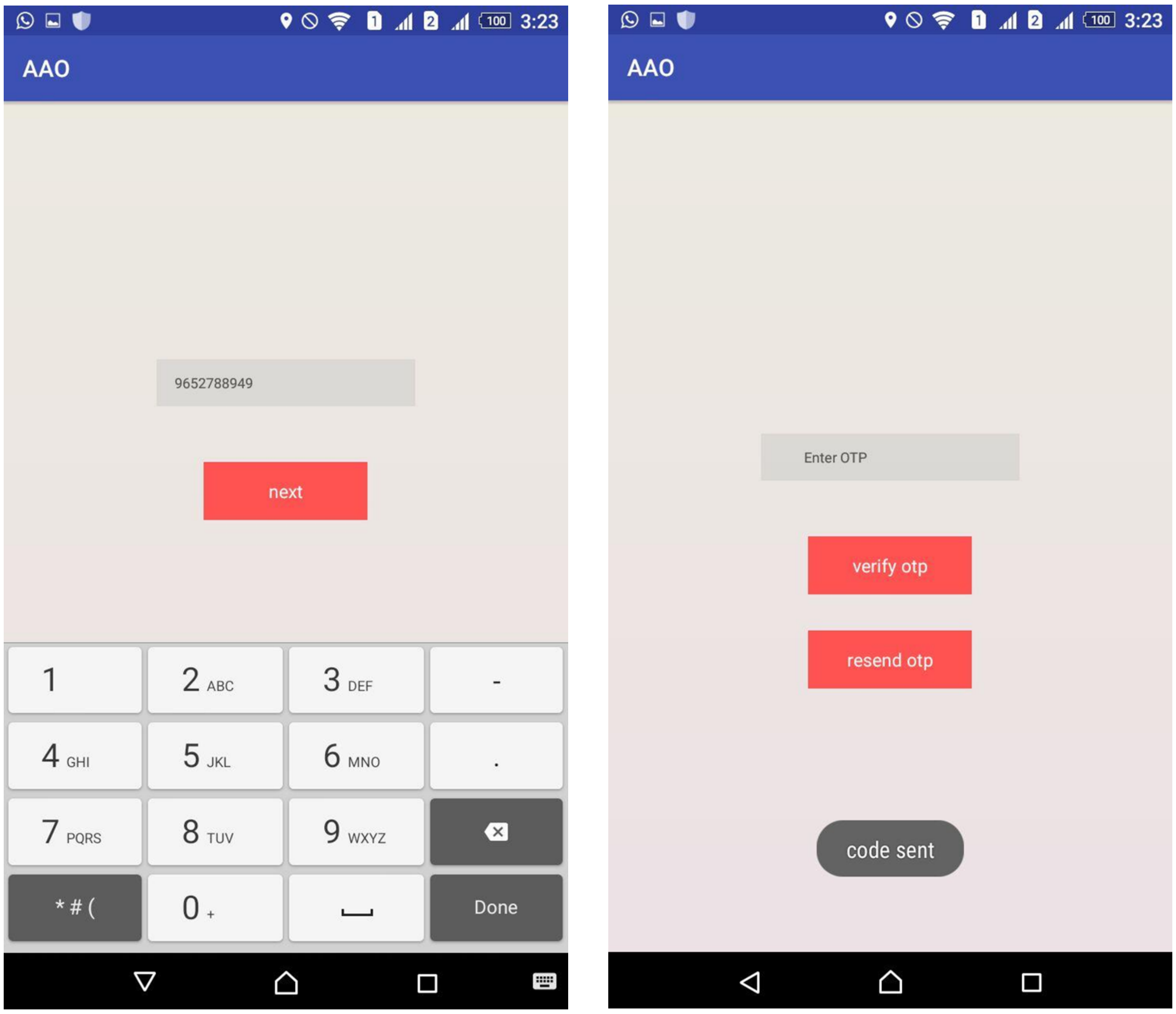


Figure 4.1: Login Screen for Artificial Agriculture Officer

The application first requires the user to enter his phone number. Then it sends a one-time-password (OTP) to the registered number. This number must be entered for final verification.

There are two ways this verification can happen:
- Instant verification: in some cases, the phone number can be instantly verified without needing to send or enter a verification code.
- Auto-retrieval: on some devices, Google Play services can automatically detect the incoming verification SMS and perform verification without user action.

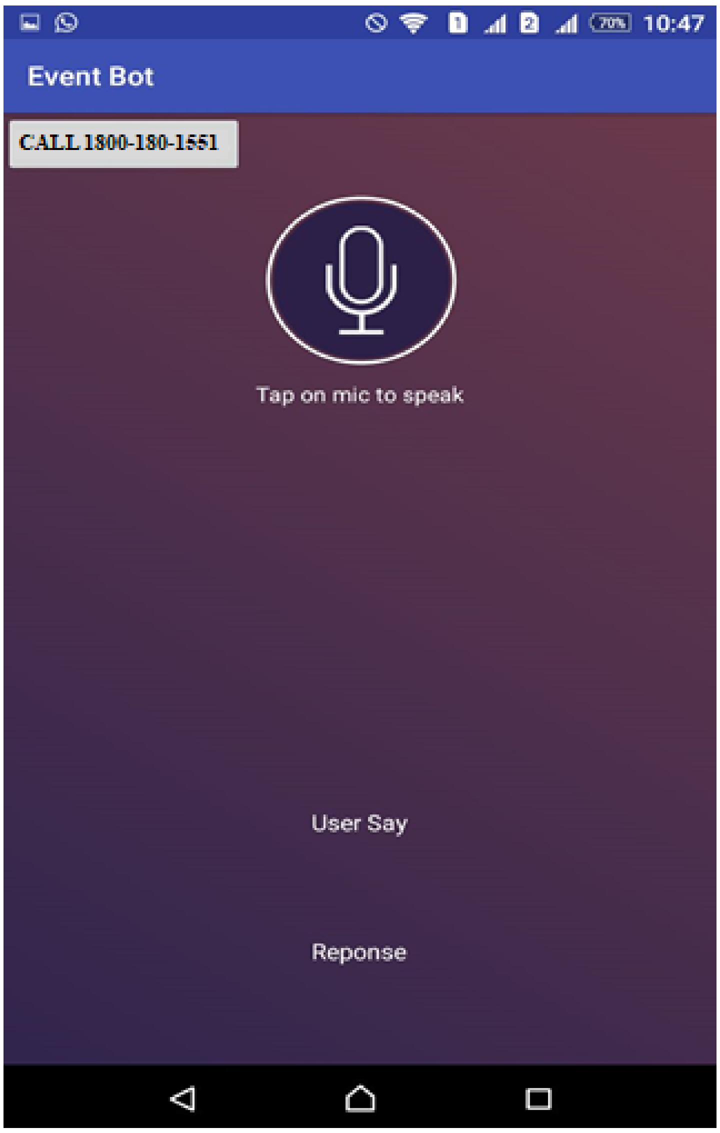
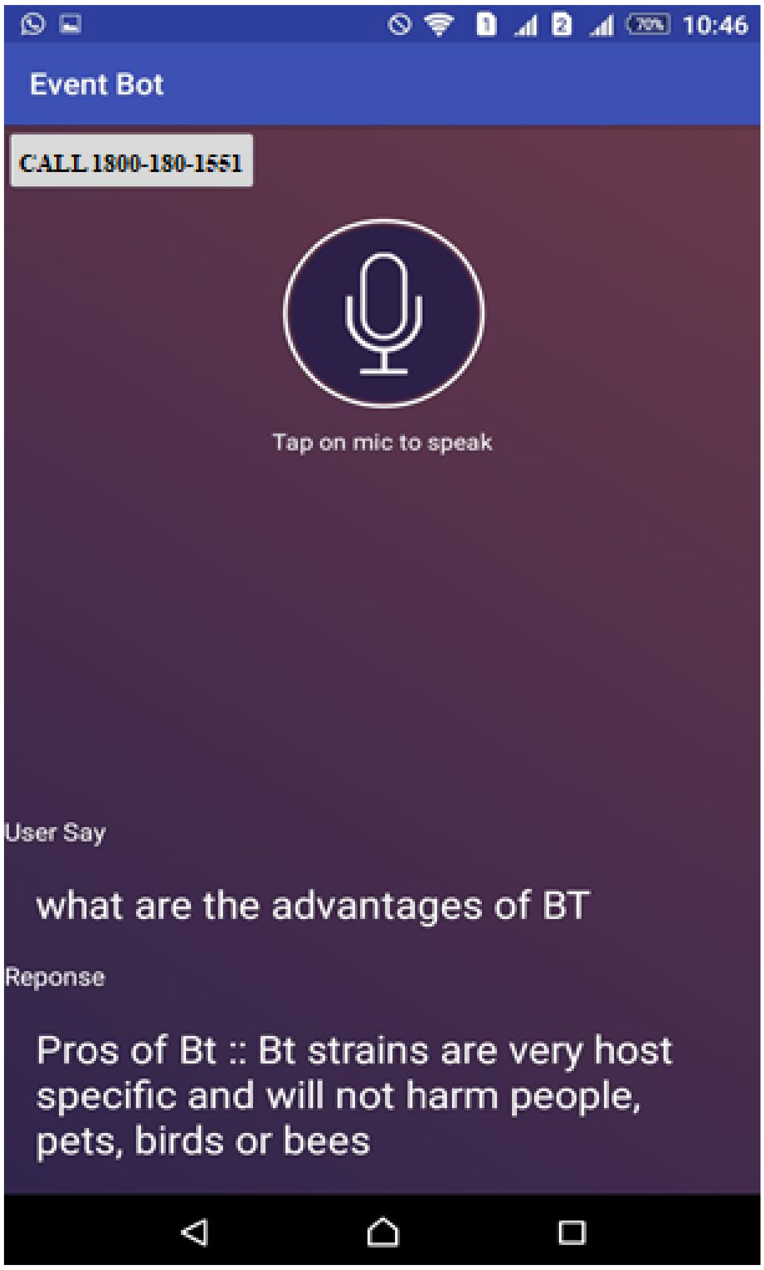


Figure 4.2: Home Screen for Artificial Agriculture Officer

The user will get the above screen upon authentication of his phone number. To ask a particular question, he will have to push the microphone button. Once the app is ready to receive an input, a pop up box will appear. At this time, the user can speak into the smartphone’s (internal or external) microphone.


10:46

Event Bot

CALL 1800-180-1551

Tap on mic to speak

User Say

what are the advantages of BT

Reponse

Pros of Bt :: Bt strains are very host specific and will not harm people, pets, birds or bees

Figure 4.3: Dialog flow for Artificial Agriculture Officer

Based on the received query, the system will search through the knowledge-base (dialogflow) and retrieve the appropriate response. Both the user's query and the response will be displayed on the screen (similar to a chat application). The response will also be converted to a speech output, and will be spoken aloud. The user can also ask follow-up questions relative to his original query. In other words, if the first question is "What is BT?", the follow up question can be "What are the advantages of it?" and the system will understand the pronoun "it" to be "BT".

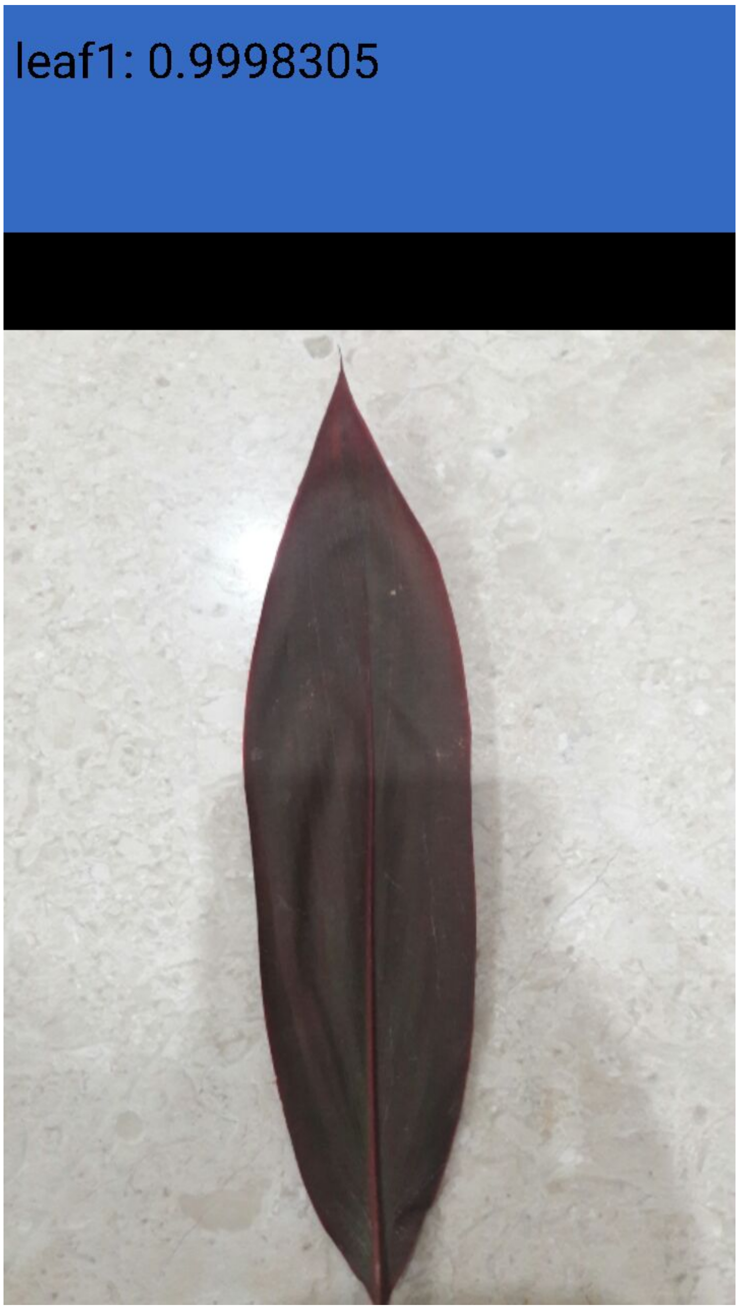

leaf1: 0.9998305

Figure 4.4: Image Recognition for Artificial Agriculture Officer

The user will get the above screen upon initiating the image recognition component. To classify a particular leaf or plant disease he will just have to aim his phone at the object. The application will identify the image and give him the classifying label at the top of the screen.

5. Conclusion and future work

Conclusion:

The application being designed is meant to be useful for agricultural workers who may not be educated to read, write, or use the internet. It also allows users to save time in searching for a particular answer. This is a better alternative to many existing systems. The app, having the speech recognition component, allows easy searching in the database. The app will be user friendly and efficient in working.

Future Work:

The application, thus far, only searches through the database created by the admins. However, there is a plan in the future to add a web scraping component as well. This will allow the application to search the internet and extract short and relevant responses to the user's queries. This idea may expand to allow the application to store these search histories so that in the future if the same question is asked, it does not have to go back to the internet. Instead it can check its stored data.

References

[1] FatihErtam, GalipAydin, “Data classification with deep learning using Tensorflow”International Conference on Computer Science and Engineering (UBMK), October 2017

[2] SharwariGaikwad, RohanAsodekar, Sunny Gadia, and Vahida Z. Attar, “AGRI-QAS question-answering system for agriculture domain,” International Conference in Advances in Computing, Comunications and Informatics (ICACCI), 2015

[3] PoojaKamavisdar , SonamSaluja, SonuAgrawal“ A Survey on Image Classification Approaches and Techniques“ International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 1, January 2013

[4] JaspreetKaur and Vishal Gupta, “Effective Question Answering Techniques and their Evaluation Metrics”, International Journal of Computer Applications (0975 – 8887) Volume 65– No.12, March 2013

[5] Sonal, Athavale, Neelabh Sao, “Classification on Moving Object Trajectories”, International Journal of Advanced Technology & Engineering Research (IJATER) ISSN No: 2250-3536 Volume 2, Issue 2, May 2012

[6] .JianxinWu,”Efficient HIK SVM Learning for Image Classification”, IEEE Transactions on Image Procecssing, Vol. 21, No. 10, October 2012

[7] Vimala.C, Dr. V. Radha, " A Review on Speech Recognition Challenges and Approaches " , World of Computer Science and Information Technology Journal (WCSIT) ISSN: 2221-0741 Vol. 2, No. 1, 1-7, 2012

[8] SantoshK.Gaikwad, BhartiW.Gawali and PravinYannawar, “A Review on Speech Recognition Technique”, International Journal of Computer Applications (0975 – 8887) Volume 10– No.3, November 2010

[9] XiaohongYu, and HongLiuHuangshan, “Image Semantic Classification Using SVM in Image Retrieval” P. R. China, 26-28, December 2009

[10] Y. Koren, “Collaborative filtering with temporal dynamics,” in Proc.KDD, pp. 447–456.Paris, France, 2009

[11] M. Chandrasekar, M. Ponnavaikko, “Tamil speech recognition: a complete model”, Electronic Journal Technical Acoustic 2008

[12] Matthias Book, Volker Gruhn, “A Notation and Framework for Dialog Flow Control in Web Applications” International Conference on Web Engineering(ICWE), 2004

Appendix

MainActivity.java

```
package com.example.hp.sttandtts;


import android.app.Activity;
import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;

import android.view.View.OnClickListener;
import android.widget.Button;
import android.view.View;
import android.widget.EditText;
import android.speech.tts.TextToSpeech;
import android.speech.tts.TextToSpeech.OnInitListener;
import android.content.Intent;
import java.util.Locale;
import android.widget.Toast;


import android.speech.RecognizerIntent;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import org.json.JSONArray;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;
import java.net.URL;
import java.net.URLConnection;
import java.util.ArrayList;

public class MainActivityextendsAppCompatActivityimplementsOnClickListener,OnInitListener {

public TextToSpeechmyTTS;
//status check code
public intMY_DATA_CHECK_CODE= 0;

private final intREQ_CODE_SPEECH_INPUT= 100;
ImageButtonbtnSpeak;
TextViewtxtSpeechInput, outputText;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.activity_main);
btnSpeak= (ImageButton) findViewById(R.id.btnSpeak);
txtSpeechInput= (TextView) findViewById(R.id.txtSpeechInput);
outputText= (TextView) findViewById(R.id.outputTex);
btnSpeak.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View view) {
promptSpeechInput();
            }
        });
        Intent checkTTSIntent = new Intent();
        checkTTSIntent.setAction(TextToSpeech.Engine.ACTION_CHECK_TTS_DATA);
startActivityForResult(checkTTSIntent, MY_DATA_CHECK_CODE);

    }

/**
     * Showing google speech input dialog
     */
private void promptSpeechInput() {

        Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
// intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
intent.putExtra(RecognizerIntent.EXTRA_PROMPT,
"Say Something");
try {
startActivityForResult(intent, REQ_CODE_SPEECH_INPUT);
        } catch (ActivityNotFoundException a) {
Toast.makeText(getApplicationContext(),
"orry! Your device doesn\\'t support speech input",
Toast.LENGTH_SHORT).show();
        }
    }

/**
     * Receiving speech input
     */
@Override
protected void onActivityResult(intrequestCode, intresultCode, Intent data) {
super.onActivityResult(requestCode, resultCode, data);

switch (requestCode) {
case REQ_CODE_SPEECH_INPUT: {
if (resultCode == RESULT_OK &&null != data) {

ArrayList<String> result = data
                            .getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                    String userQuery = result.get(0);
txtSpeechInput.setText(userQuery);
RetrieveFeedTask task = new RetrieveFeedTask();
task.execute(userQuery);


                }
break;
            }
        }
if (requestCode == MY_DATA_CHECK_CODE) {
//the user has the necessary data - create the TTS
if (resultCode == TextToSpeech.Engine.CHECK_VOICE_DATA_PASS)
myTTS= new TextToSpeech(this, this);
else {
//no data - install it now
Intent installTTSIntent = new Intent();
                installTTSIntent.setAction(TextToSpeech.Engine.ACTION_INSTALL_TTS_DATA);
startActivity(installTTSIntent);
            }
        }
    }


// Create GetTextMetod
public String GetText(String query) throws UnsupportedEncodingException {

        String text = "";
BufferedReader reader = null;

// Send data
try {

// Defined URL  where to send data
URL url = new URL("https://api.dialogflow.com/v1/query?v=20150910");

// Send POST data request

URLConnection conn = url.openConnection();
conn.setDoOutput(true);
conn.setDoInput(true);

conn.setRequestProperty("Authorization", "Bearer 34af6c0255814d509322ead6e282b649");
conn.setRequestProperty("Content-Type", "application/json");

//Create JSONObject here
JSONObjectjsonParam = new JSONObject();
JSONArrayqueryArray = new JSONArray();
queryArray.put(query);
jsonParam.put("query", queryArray);
//            jsonParam.put("name", "order a medium pizza");
jsonParam.put("lang", "en");
jsonParam.put("sessionId", "1234567890");


OutputStreamWriterwr = new OutputStreamWriter(conn.getOutputStream());
Log.d("karma", "after conversion is " + jsonParam.toString());
wr.write(jsonParam.toString());
wr.flush();
Log.d("karma", "json is " + jsonParam);

// Get the server response
```

```
reader = new BufferedReader(new InputStreamReader(conn.getInputStream()));
StringBuildersb = new StringBuilder();
            String line = null;


// Read Server Response
while ((line = reader.readLine()) != null) {
// Append server response in string
sb.append(line + "\n");
            }


            text = sb.toString();


JSONObject object1 = new JSONObject(text);
JSONObject object = object1.getJSONObject("result");
JSONObjectfulfillment = null;
            String speech = null;
//            if (object.has("fulfillment")) {
fulfillment = object.getJSONObject("fulfillment");
//                if (fulfillment.has("speech")) {
speech = fulfillment.optString("speech");
//                }
//            }


Log.d("karma ", "response is " + text);
return speech;

        } catch (Exception ex) {
Log.d("karma", "exception at last " + ex);
        } finally {
try {

reader.close();
            } catch (Exception ex) {
            }
        }

return null;
    }

@Override
public void onClick(View view) {
    }

@Override
public void onInit(intinitStatus) {
if (initStatus == TextToSpeech.SUCCESS) {
if (myTTS.isLanguageAvailable(Locale.US) == TextToSpeech.LANG_AVAILABLE)
myTTS.setLanguage(Locale.US);
        } else if (initStatus == TextToSpeech.ERROR) {
Toast.makeText(this, "Sorry! Text To Speech failed...", Toast.LENGTH_LONG).show();
        }
    }
```

```
class RetrieveFeedTaskextendsAsyncTask<String, Void, String> {

@Override
protected String doInBackground(String... voids) {
            String s = null;
try {

                s = GetText(voids[0]);


            } catch (UnsupportedEncodingException e) {
e.printStackTrace();
Log.d("karma", "Exception occurred " + e);
            }

return s;
        }

@Override
protected void onPostExecute(String s) {
super.onPostExecute(s);
outputText.setText(s);
            String words = s.toString();
speakWords(words);

        }

private void speakWords(String speech) {

//speak straight away
myTTS.speak(speech, TextToSpeech.QUEUE_FLUSH, null);
        }

    }
}
```