

**Plant Disease Detection and
Prescription.**

**Detail Solution Design &
Architecture.**

Table of Contents

1. Introduction	3
2. System Description	4
2.1 Technical Architecture	4
3. Factors Influencing Design	9
3.1 Assumptions and Dependencies	9
3.2 Constraints	10

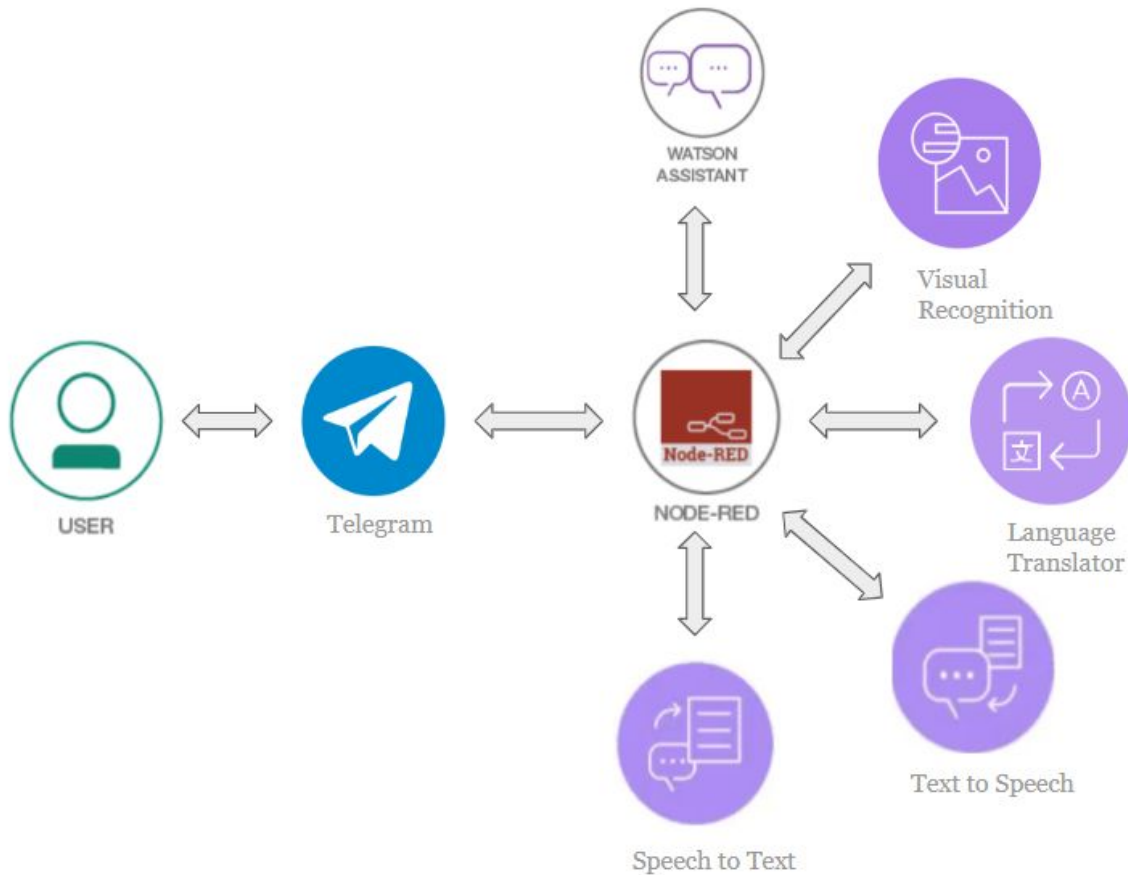
1. Introduction

Plant diseases and pests are widely considered an important cause of crop losses, and sometimes a threat to the food supply. People not being able to recognize the disease in the early stages due to which the spread of the disease increases. We are losing a significant amount of yield on a global scale to pests and diseases, at a time when we must increase production to feed a growing population.

Our solution to early disease detection is based on an **artificial neural network**, which is now the most robust technique for image classification. The main advantages of our solution include **high processing speed** and **high classification accuracy**. Our solution model takes images as input and predicts the disease accordingly. Our solution model recognizes particular plant diseases. However, our further research is to make a plant disease recognition model that can work as a universal detector, recognizing general abnormalities on the leaves, such as scorching or mold. And we also created a **chatbot** that prescribes the **remedy** that can be taken against the **disease**. This whole service can be used by anyone through ‘ Telegram ’ by simply uploading images of affected leaves by which we can get the disease and some symptoms, cause, and remedies(management) regarding the disease. If a user just wants to get some remedies regarding plant disease, he/she can just give the plant and its disease name to get the remedies.

2. System Description

2.1 Technical Architecture

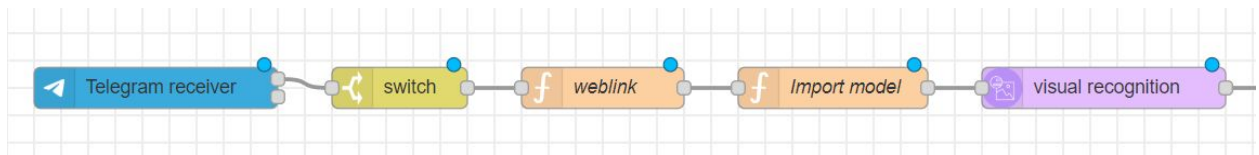


Our solution architecture consists of the integration of following IBM Watson services and **TELEGRAM** with **NODE RED**,

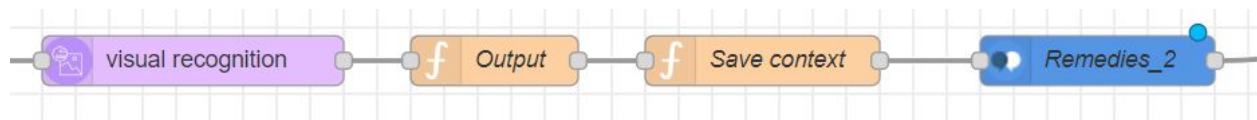
- Visual Recognition Service
- Watson Assistant Service
- Language Translator Service
- Text to Speech Service
- Speech to Text Service

INTEGRATIONS WITH NODE-RED:

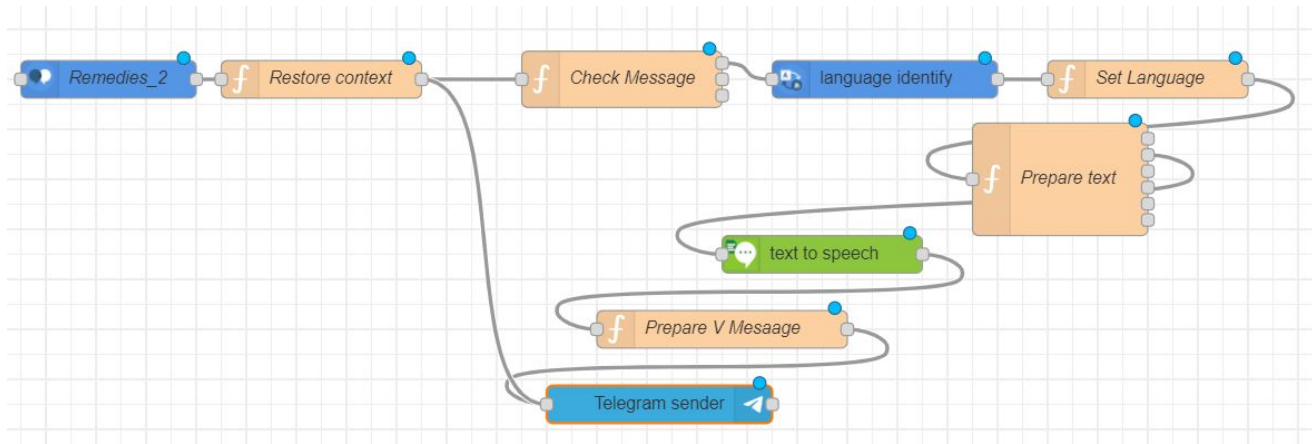
- Basically, to integrate a telegram chatbot, we need two telegram nodes, a **telegram receiver**, and a **telegram sender**.
 - The telegram receiver node retrieves the query of the user.
 - The telegram sender node gives back the reply to the user.
- **Telegram receiver** is connected to our **visual recognition service** through the following nodes:



- Then, the **visual recognition service** is connected to one of our created **Watson assistant(Remedies_2)** chatbots through the following nodes:



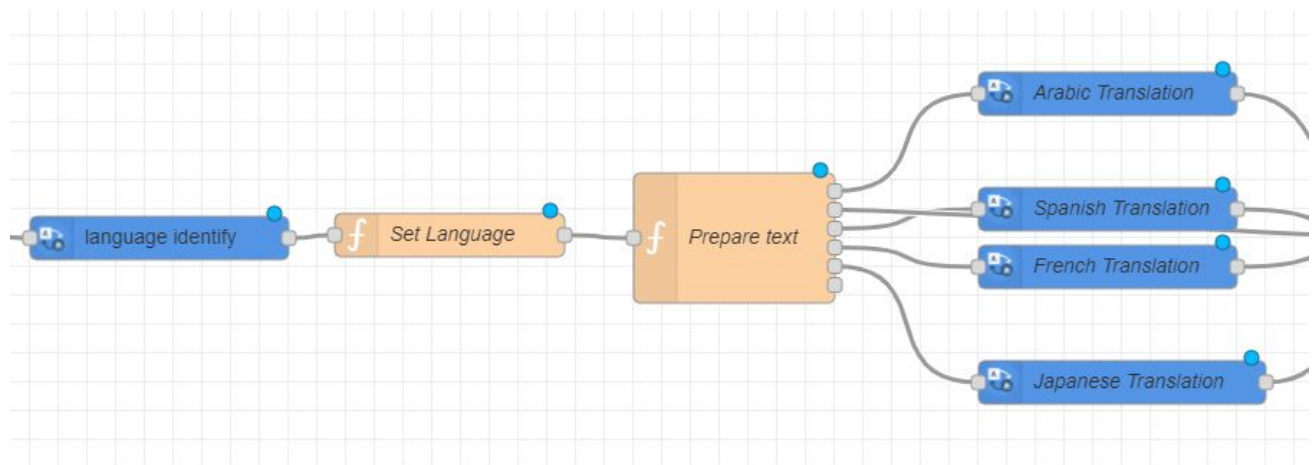
- Then, this **Watson assistant** is connected to **telegram sender** in the following ways,
 - **WATSON ASSISTANT ⇒ LANGUAGE IDENTIFY NODE(Uses Language translator Service) ⇒ TEXT TO SPEECH SERVICE ⇒ TELEGRAM SENDER.**
This step gives out the voice message.
 - **WATSON ASSISTANT ⇒ TELEGRAM SENDER.**
This step gives out the text message.



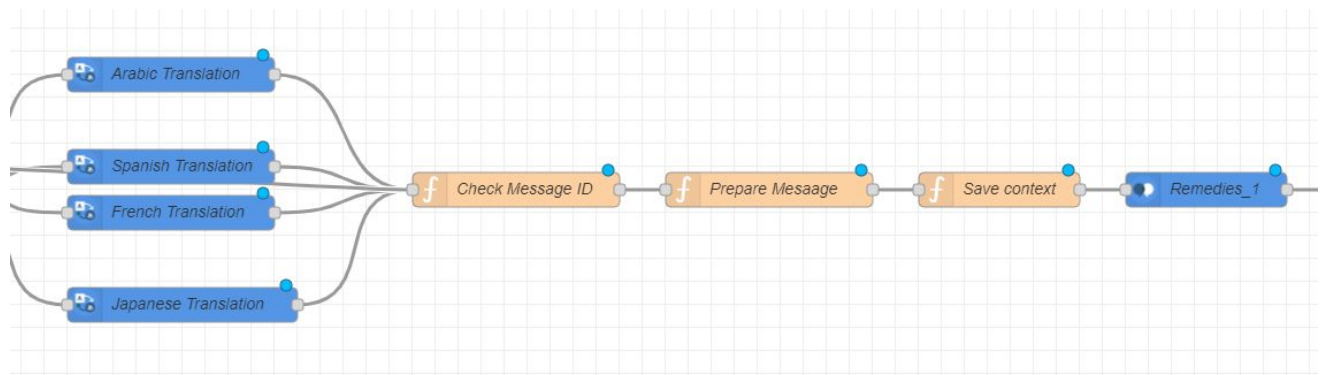
- The above flow of connections works when a user wants to know the disease and remedies by uploading the image. But if a user wants to get remedies by giving the plant name and its disease name(This is the case if the user just want to get some information about plant diseases and remedies), then the following flow of connections comes into the picture,
- **Telegram receiver** is connected to the **language identify** node through the 'check message' functional node.



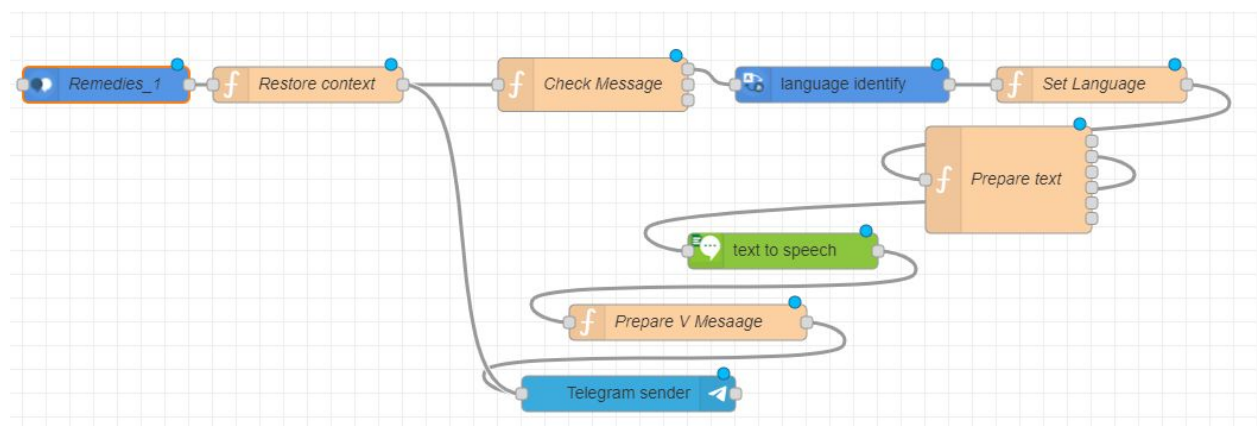
- **Language identify** node is connected to four **language translator** nodes(In this case, the user can text in four other languages(Spanish, French, Japanese, and Arabic) also other than English for which these four language translator nodes are used. However, the reply from the bot will be in English.



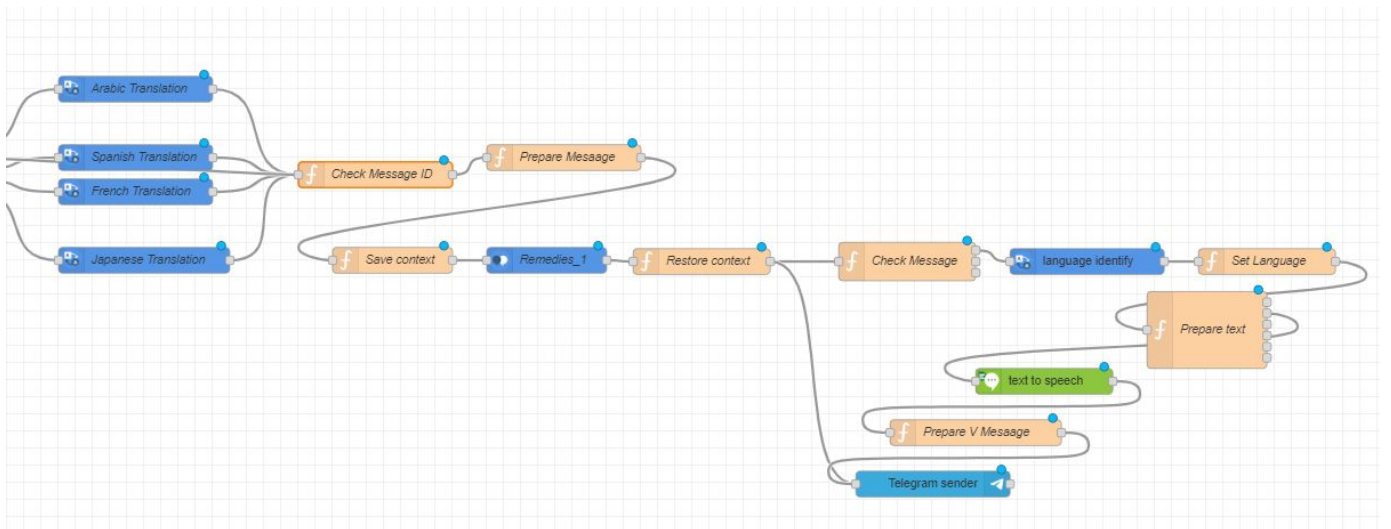
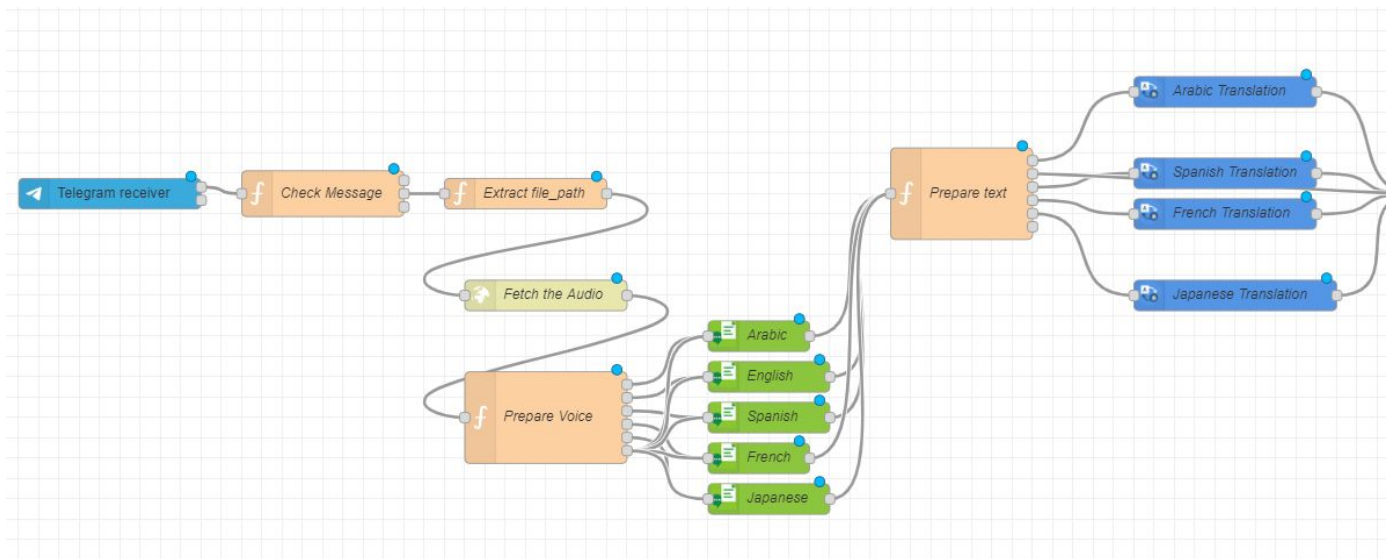
- Then, the **language translator** nodes are connected to our second **Watson assistant(Remedies_1)**.



- And, this **Watson assistant** is connected to the **telegram sender**(One link is through **Text to Speech Service** node to get the voice message and another one normally to get the text message).



- And, users can also use voice messages to ask for remedies. For this reason, the same **telegram receiver** is connected to five **Speech to Text Service** nodes and then to the **language translator** nodes which is then connected to the same **Watson assistant**, finally to **telegram sender**(One link is through **Text to Speech Service** node to get the voice message and another one normally to get the text message).



3. Factors Influencing Design

This section describes the standards, assumptions, and constraints that influence the technical design of the proposed system.

3.1 Assumptions and Dependencies

Assumptions:

- The user should have a telegram account.
- The user using this telegram bot should understand the English language.

Dependencies:

- **node-red-contrib-telegrambot** : This package contains a receiver and a sender node that are used to construct a Telegram Bot. The only thing required is the token that can be retrieved by the **@botfather** Telegram Bot.
- **node-red-node-watson** : A collection of nodes to interact with the IBM Watson services in IBM Cloud. (Nodes like assistant v2, visual recognition, text to speech, speech to text, etc..)
- **node-red** : The Node-RED palette includes a default set of nodes that are the basic building blocks for creating flows. (Nodes like function, switch, etc..)

3.2 Constraints

Constraints imposed because of the dataset we used for training our model:

- This solution model works well if the input test images are 256×256 images (nearer to 256×256 dimension images), as the model is trained with 256×256 dimension images while training.
- For now, the model recognizes only 26 plant diseases and 12 healthy plants. After extensive training on diverse datasets, our model will be capable of distinguishing a large number of different diseases.
- Another limitation is that currently, the proposed method cannot detect multiple diseases in one image.
- The model may not predict efficiently if the input image has 2 leaves in it. (As the images for training contain only one leaf.)