CONTENTS

Acknowledgements

It is our proud privilege and duty to acknowledge the kind of guidance and help received from several people in preparation of this project. It would not have been possible to accomplish this task without their valuable help, cooperation and guidance.

We convey thanks to our guide Dr. Deepa Deshpande, Asst. Professor (CSE) for providing encouragement, constant support and guidance which was instrumental in the completion of this project successfully.

We are grateful to Dr. Vijaya Musande, Head of the Department (CSE) for giving us the support and encouragement along with all available provisions in our department to ensure that our project work went along smoothly.

We would also like to thank to Dr. H.H.Shinde, Principal of Jawaharlal Nehru Engineering College Aurangabad for his constant watch over our progress and for backing us along every step of the way.

We would also like to thank Mr. Kedar Dixit, our mentor assigned by Persistent Systems Ltd. for guiding us through the development of our project ensuring that we are keeping up to the industry standards in terms of quality of the software. It was an honour to work under a person with such knowledge about the software industry, since it helped get an insight into corporate life.

Last but not the least we would like to thank all our family and friends for keeping us motivated and believing us throughout the course of this project.

Abstract

India being a country mainly based on agricultural revenue, small and midrange farm owners face the threat of pests and crop diseases on a daily basis. This in turn reduces the total yield of crops by a significant percentage. Many such farmers misdiagnose the disease that has actually infested their farm which causes them to take the wrong measures of damage control. In recent years, cotton farmers in Maharashtra have been affected by the Pink Bollworm and have faced massive losses, which is just one example of the current scenario troubling the farming industry in India.

The technological solution will use a deep learning approach that will assist farmers to alleviate the infestation by identifying the insect or crop disease from simple photos in real time and recommend corrective measures for their eradication.

A convolutional neural network has been trained using a dataset of insect and crop disease images for the image based classification, which is an implementation of supervised learning.

With the rise of the internet and mobile technology worldwide, information is at our fingertips. Also, smartphones with powerful cameras can help to scale up our solution of insect or crop disease identification to a feasible and practical extent.

Our technological solution is an Android application, enabling the user to capture or select an existing image of an insect or disease affecting the crop and upload it. These 2D images will act as an input to the trained model. The image will then be classified which will conclude the identification process. The result returned will consist of top three predictions with necessary information of the insects or crop diseases identified:

- Common Name
- Scientific Name
- Damage
- Distribution
- Preventive measures

1. Introduction

1.1 Introduction

Computer vision is an interdisciplinary field that deals with how computers can be made for gaining high-level understanding from digital images or videos. From the perspective of engineering, it seeks to automate tasks that the human visual system can do.

Computer vision tasks include methods for acquiring, processing, analysing and understanding digital images, and extraction of high-dimensional data from the real world in order to produce numerical or symbolic information, e.g., in the forms of decisions. Understanding in this context means the transformation of visual images (the input of the retina) into descriptions of the world that can interface with other thought processes and elicit appropriate action. This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory.

As a scientific discipline, computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, or multi-dimensional data from a medical scanner. As a technological discipline, computer vision seeks to apply its theories and models for the construction of computer vision systems.

Sub-domains of computer vision include scene reconstruction, event detection, video tracking, object recognition, 3D pose estimation, learning, indexing, motion estimation, and image restoration.

The classical problem in computer vision, image processing, and machine vision is that of determining whether or not the image data contains some specific object, feature, or activity. Different varieties of the recognition problem are described in the literature.

Object recognition (also called object classification) – one or several pre-specified or learned objects or object classes can be recognized, usually together with their 2D positions in the image or 3D poses in the scene. Blippar, Google Goggles and LikeThat provide stand-alone programs that illustrate this functionality.

Identification – an individual instance of an object is recognized. Examples include identification of a specific person's face or fingerprint, identification of handwritten digits, or identification of a specific vehicle.

Detection – the image data are scanned for a specific condition. Examples include detection of possible abnormal cells or tissues in medical images or detection of a vehicle in an automatic road toll system. Detection based on relatively simple and fast computations is sometimes used for finding smaller regions of interesting image data which can be further analysed by more computationally demanding techniques to produce a correct interpretation.

Currently, the best algorithms for such tasks are based on convolutional neural networks. An illustration of their capabilities is given by the ImageNet Large Scale Visual Recognition Challenge; this is a benchmark in object classification and detection, with millions of images and hundreds of object classes. Performance of convolutional neural networks, on the ImageNet tests, is now close to that of humans.

One of the major areas Computer Vision can be applied is in image classification and object identification. This application displays the versatility of computer vision because using computational methods we would be able to automate tasks with relatively high accuracy which previously required substantial human interference. Various areas of our society can benefit from such a technological advancement like:

- Healthcare, to identify the existence of diseases on human body by analysing reports or images of the infected regions.
- Law Enforcement, to nab criminals by automatic identification on CCTV footage or image captures from integrated monitoring systems.
- Administration, to help identification processes in governmental institutions like Certificate Issuing Authorities by automation.
- Security, to aid security teams at high-profile locations like Airports with analysis of luggage and reduce time-lag of passenger screening.
- Education, to go digital by translating all handwritten documents into digital copies using OCR to help easier and eco-friendly distribution of books.

Keeping such examples in mind it came to our attention that Agriculture is one such domain that can hugely benefit from the aid of Computer Vision, specifically object recognition and image classification. If farmers can be equipped with applications to tackle various day to day problems they face, it will play a major role in pushing the agriculture industry back onto the right track of growth and development. This is how the idea behind the project came about as explained below.

1.2 Necessity

Upon correspondence with various experts and novices in the field of agricultural education, a BSc. Undergraduate student put forth a scenario which could be optimised using image classification. The scenario was such that farmers who needed advice from agriculture experts to subject their farm to dosages of various pesticides and insecticides had to go through the tedious task of requesting the said expert to look into the issue remotely, analyse it and then recommend the remedy for the same. The biggest drawback in this is the amount of time it would take for the solution to reach the farmer and then subsequently for them to administer the recommended dosage of the control measures advised.

Our team concluded that we could cut the time delay to a matter of seconds if we use Computer Vision, specifically object recognition and image classification to provide a solution with relatively high accuracy to the farmer without any need of instantly communicating with an expert, which can be avoided unless absolutely necessary.

1.3 Image Classification

Object Classification is an important task within the field of computer vision. Image classification refers to the labelling of images into one of a number of predefined categories. Classification includes image sensors, image pre-processing, object detection, object segmentation, feature extraction and object classification. Many classification techniques have been developed for image classification. In this survey various classification techniques are considered; Artificial Neural Network (ANN), Decision Tree (DT), Support Vector Machine (SVM) and Fuzzy Classification.

1.4 Approaches to Image Classification

Artificial Neural network ANN is a type of artificial intelligence that imitates some functions of the person mind. ANN has a normal tendency for storing experiential knowledge. An ANN consists of a sequence of layers, each layer consists of a set of neurones. All neurones of every layer are linked by weighted connections to all neurones on the preceding and succeeding layers. It uses Non-parametric approach. Performance and accuracy depends upon the network structure and number of inputs.

Decision Tree calculates class membership by repeatedly partitioning a dataset into uniform subsets Hierarchical classifier permits the acceptations and rejection of class labels at each intermediary stage. This method consists of 3 parts: Partitioning the nodes, find the terminal nodes and allocation of class label to terminal nodes Decision Tree are based on hierarchical rule based method and use Non-parametric approach.

A support vector machine builds a hyper plane or set of hyper planes in a high- or infinite dimensional space, used for classification. Good separation is achieved by the hyper plane that has the largest distance to the nearest training data point of any class (functional margin), generally larger the margin lower the generalization error of the classifier. SVM uses Nonparametric with binary classifier approach and can handle more input data very efficiently. Performance and accuracy depends upon the hyperplane selection and kernel parameter.

In Fuzzy classification, various stochastic associations are determined to describe characteristics of an image. The various types of stochastic are combined (set of properties) in which the members of this set of properties are fuzzy in nature. It provides the opportunity to describe different categories of stochastic characteristics in the similar form. It uses stochastic approach. Performance and accuracy depends upon the threshold selection and fuzzy integral.

1.5 Why Convolutional Neural Network?

Current approaches to object recognition make essential use of machine learning methods. To improve their performance, we can collect larger datasets, learn more powerful models, and use better techniques for preventing overfitting.

To learn about numerous objects from thousands of images, a model with a large learning capacity is required. However, the immense complexity of the object identification task means that this problem cannot be specified even by a dataset as large as ImageNet, so our model should also have lots of prior knowledge to compensate for all the data we don't have. Convolutional neural networks (CNNs) constitute one such class of models [16, 11, 13, 18, 15, 22, and 26]. Their capacity can be controlled by varying their depth and breadth, and they also make strong and mostly correct assumptions about the nature of images (namely, stationarity of statistics and locality of pixel dependencies). Thus, compared to standard feedforward neural networks with similarly-sized layers, CNNs have much fewer connections and parameters and so they are easier to train, while their theoretically-best performance is likely to be only slightly worse.

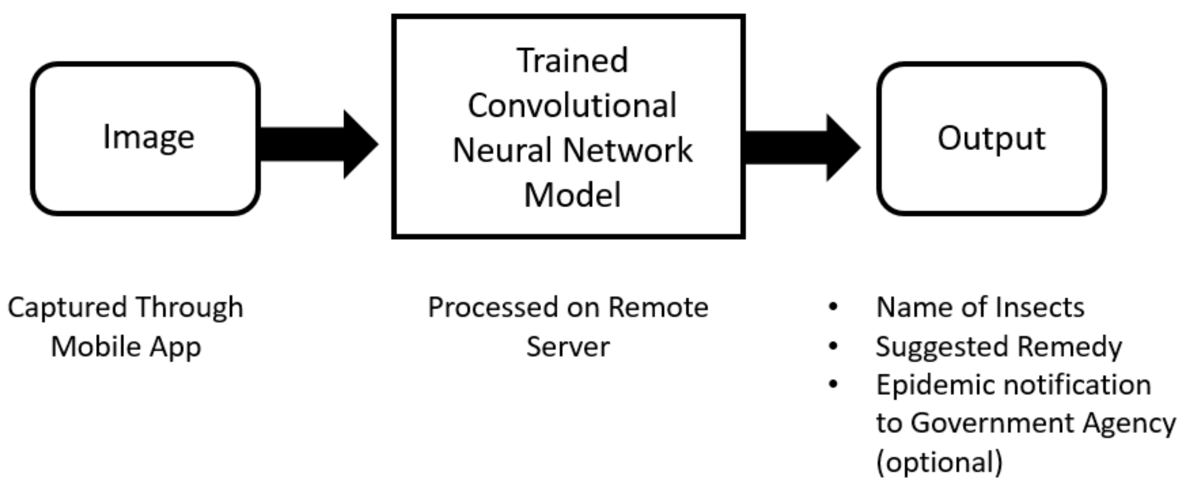


Figure 1 Block Diagram of System

2. Literature Survey

A literature survey is a summarization of the technical articles and research papers referred to, which includes the current knowledge including substantive findings, as well as theoretical and methodological contributions to a particular topic which is a part of the software development process. Listed below are the works of some researchers we have used to develop this system.

A paper by Oonagh N Bryne [1], helped us understand that introduction of high yielding varieties, expansion in irrigation facilities and indiscriminate use of increased rates of agrochemicals such as fertilizers and pesticides in recent years with a view to increase productivity has resulted in heavy crop losses due to insect pests in certain crops. This situation has risen mainly due to elimination of natural enemies, resurgence of pests, and development of insecticide resistance and out-break of secondary pests. Distribution, nature of damage, life history of important key pests of crops and their management strategies were outlined as a part of the research.

The research paper by Alex Krizhevsky [2] stated that a large, deep convolutional neural network is capable of achieving record breaking results on a highly challenging dataset using purely supervised learning and that a network's performance degrades if a single convolutional layer is removed.

Jia Deng's paper [3], stated that ImageNet could be used as a training resource, a benchmark dataset, to introduce new semantic relations for visual modelling and for Human vision research.

A paper by Dunlu Peng et al. [4] was a motivation for using JSON. Some highlights were the comparison between XML and JSON. It stated that the default standard used for data exchange in web service applications is XML. XML data has to be parsed at both client and server end as it semi-structured. This parsing consumes time and memory. As compared to XML, JSON is lightweight key valued style data exchanging format. Data Advanced Binding is used to for processing JSON which maps JSON data onto the host language by dynamically building Data Mapping Template (DMT).

We also read about android app development and security issues from a paper by Suhas Holla and Mahima M Katti [5]. It stated that development in mobile application is at meteor pace which is giving user a rich and fast. Android released by Google is an open source mobile phone operating system with Linux-based platform. Operating system, middleware, and user interface and application software together from Android. Android is the currently used OS in mobile phones. Android is vulnerable to criminal attacks. For the purpose Android Application Sandbox (AASandbox) perform static and dynamic analysis for Android programs to detect suspicious applications automatically.

Another interesting read was a paper by Amarpreet Singh Johal and Baljit Singh [6]. The goal of this paper was to analyze the performance of SOAP and RESTful web services by comparing the two approaches using the different methods like GET, POST, PUT, and DELETE. Response time for all four methods of a particular service is evaluated for both REST based and SOAP based Web services. REST outperforms SOAP in all the requests as it provides high performance and perfectly good solution for the majority of implementations on mobile devices. So we decided to choose RESTful based Web service instead of SOAP based Web service for communication between mobile and API.

In the study by B. Jaya Kaviya [7], we read that web services composition has been implemented for combining the various web services as used in many field. It states that the RESTful services has greater advantage then SOAP .This supports scalability as it is stateless which allow to include additional server behind a load balancer. The Uniform interface allows to document which is independent of the operation of API and it is defined to be simplicity. The Restful services established to support the navigation between the services in the selection of services. Even though many services has been integrated at the server, it has the appearance of single service to the user. The probability of satisfying the user demands is high in these services composition.

The paper by Supriya S. Pore, Swalaya B. Pawar [8], helped us decide upon using NoSQL database for our project. It had interesting insights comparing the traditional SQL databases with NoSQL databases. The main aim of the research paper was to evaluate the basics of SQL and NoSQL databases and the comparative analysis of these two databases. It also described the Axiomatic of SQL and NoSQL databases.

3. System Development

3.1 Shortlisting of Harmful Insects and Crop Diseases

The first phase of work was shortlisting of the insects and crop diseases for which our application would provide identification support. This process involved sifting through various researches by agriculture experts and entomologists whose studies were detrimental in stating whether certain insects do cause harm to crops and what crop diseases are widespread in regions of India where common crops like rice, wheat, cotton etc. are grown. We also had to ensure that images of these insects and crop diseases are available online in abundance, and that their features are distinguishable to help accurate identification. After thorough research we shortlisted a set of 15 insects. We also came across a readymade dataset of crop diseases used for a project called Plant Village, which was available on GitHub. Hence we decided upon 15 insect and 37 crop disease classes for training our model on. The classes are listed in the Performance Analysis section.

3.2 Data acquisition

The most important and crucial part of an image classification model is the data that it is trained upon. It is essential that the data is diverse and available in large quantities. The model needs to be trained on data that ranges from high quality, high resolution images to ones that are not as perfect. If the training images can replicate what a user might click from their camera in real-time it gives the model the best chance to identify the image on runtime, since it has been prepared for such use cases.

In order to introduce such variety and diversity in the datasets used for our model, different sources were shortlisted:

- Web Scraping
- ImageNet
- Existing Project Repositories

The final datasets created are a cumulative sum of all these sources which have helped introduce the required noise as well as ensure that the model has enough data to learn and conclude weights from.

3.2.1 Web Scraping

Web scraping is a computer software technique of extracting information from websites. This technique mostly focuses on the transformation of unstructured data (HTML format) on the web into structured data (database or spreadsheet).

The need and importance of extracting data from the web is becoming increasingly loud and clear. For example, creating an index of interest and sentiment about various data science courses available on the internet. This would not only require finding out new courses, but also scrape the web for their reviews and then summarizing them in a few metrics. This is one of the problems / products, whose efficacy depends more on web scrapping and information extraction (data collection) than the techniques used to summarize the data.

There are several ways to extract information from the web. Use of APIs being probably the best way to extract data from a website. Almost all large websites like Twitter, Facebook, Google, Twitter, and Stack Overflow provide APIs to access their data in a more structured manner. If an API is available, it is almost always the preferred approach over web scraping. This is because access to structured data from the provider is available, hence there is no need to create an engine to extract the same information.

Not all websites provide an API. Some do it because they do not want the readers to extract huge information in structured way, while others don’t provide APIs due to lack of technical knowledge. What do you do in these cases? Well, we need to scrape the website to fetch the information. There might be a few other ways like RSS feeds, but they are limited in their use. A non-programming way to extract information out of web pages is ‘import.io’. It provides a GUI driven interface to perform all basic web scraping operations.

Scraping can be implemented in various ways, including use of Google Docs to almost every programming language. Python is one of the best options because of its ease and rich ecosystem. It has a library known as ‘BeautifulSoup’ which assists this task.

Python is an open source programming language. It provides many libraries to perform one function. Hence, it is necessary to find the best to use library. BeautifulSoup (python library), was preferred since it is easy and intuitive to work on. Precisely, two Python modules have been used for scraping data:

- Urllib2: It is a Python module which can be used for fetching URLs. It defines functions and classes to help with URL actions (basic and digest authentication, redirections, cookies etc.). For more detail refer to the documentation page.
- BeautifulSoup: It is an incredible tool for pulling out information from a webpage. You can use it to extract tables, lists, paragraph and you can also put filters to extract information from web pages. In this article, we will use latest version BeautifulSoup 4. You can look at the installation instruction in its documentation page.

BeautifulSoup does not fetch the web page for us. That’s why, urllib2 has been used in combination with the BeautifulSoup library.

Python has several other options for HTML scraping in addition to BeautifulSoup. Here are some others:

- mechanize
- scrapemark
- scrapy

In the web scraping phase of data acquisition we were able to download image in batches of around 100-150 for each class. These datasets were then cleaned since they also included unnecessary images. Upon cleaning the size was further narrowed down to 80-100 images of each class.

This meant that more data had to be collected before training began, hence other options were explored.

3.2.2 ImageNet

ImageNet is a dataset of over 15 million labelled high-resolution images belonging to roughly 22,000 categories. The images were collected from the web and labelled by human labellers using Amazon’s Mechanical Turk crowd-sourcing tool. Starting in 2010, as part of the Pascal Visual Object Challenge, an annual competition called the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) has been held. ILSVRC uses a subset of ImageNet with roughly 1000 images in each of 1000 categories. In all, there are roughly 1.2 million training images, 50,000 validation images, and 150,000 testing images.

ImageNet consists of variable-resolution images, while our system requires a constant input dimensionality. Therefore, we down-sampled the images to a fixed resolution of 256 × 256. Given a rectangular image, we first rescaled the image such that the shorter side was of length 256, and then cropped out the central 256×256 patch from the resulting image. We did not pre-process the images in any other way, except for subtracting the mean activity over the training set from each pixel. So we trained our network on the (centred) raw RGB values of the pixels.

3.2.3 Existing Project Repositories

Plant Village Master is a readily available repository for plant disease images. This dataset provided sufficient images for all the crop disease classes along with raw, grayscale, colour and segmented variety of each individual image.

After accumulation we collected roughly 4500 images of 15 insect classes and 54,000 images of 38 crop disease classes.

3.3 Data Augmentation

Deep networks need large amount of training data to achieve good performance. To build a powerful image classifier using very little training data, image augmentation is usually required to boost the performance of deep networks. Image augmentation artificially creates training images through different ways of processing or combination of multiple processing, such as random rotation, shifts, shear and flips, etc. In this system we used augmentation to create 15 copies of each original image.

3.3.1 Average Filter

The idea of mean filtering is simply to replace each pixel value in an image with the mean (`average') value of its neighbours, including itself. This has the effect of eliminating pixel values which are unrepresentative of their surroundings. Mean filtering is usually thought of as a convolution filter. Like other convolutions it is based around a kernel, which represents the shape and size of the neighbourhood to be sampled when calculating the mean. In our case we used this filter to provide a blurring effect to the original which resulted in a completely new training example.

3.3.2 Gaussian Filter

In image processing, a Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. The visual effect of this blurring technique is a smooth blur resembling that of viewing the image through a translucent screen, distinctly different from the bokeh effect produced by an out-of-focus lens or the shadow of an object under usual illumination. Gaussian smoothing is also used as a pre-processing stage in computer vision algorithms in order to enhance image structures at different scales or image augmentation.

Mathematically, applying a Gaussian blur to an image is the same as convolving the image with a Gaussian function. This is also known as a two-dimensional Weierstrass transform. Since the Fourier transform of a Gaussian is another Gaussian, applying a Gaussian blur has the effect of reducing the image's high-frequency components.

The Gaussian function for 2 dimensions can be represented as

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

3.3.3 Cropping

In our case we used 5 different variations of cropping for every training example.

The variations are

1. Cropping from 4 different corners
2. Cropping at the centre.

3.3.4 RGB to Grey

A grayscale (or gravy level) image is simply one in which the only colors are shades of gray. The reason for differentiating such images from any other sort of color image is that less information needs to be provided for each pixel. In fact a `gray' color is one in which the red, green and blue components all have equal intensity in RGB space, and so it is only necessary to specify a single intensity value for each pixel, as opposed to the three intensities needed to specify each pixel in a full color image.

Often, the grayscale intensity is stored as an 8-bit integer giving 256 possible different shades of gray from black to white. If the levels are evenly spaced then the difference between successive gray levels is significantly better than the gray level resolving power of the human eye.

Grayscale images are very common, in part because much of today's display and image capture hardware can only support 8-bit images. In addition, grayscale images are entirely sufficient for many tasks and so there is no need to use more complicated and harder-to-process colour images.

3.3.5 Rotation

The rotation operator performs a geometric transform which maps the position (x_1, y_1) of a picture element in an input image onto a position (x_2, y_2) in an output image by rotating it through a user-specified angle **Θ** about an origin O In most implementations, output locations (x_2, y_2) which are outside the boundary of the image are ignored. Rotation is most commonly used to improve the visual appearance of an image, although it can be useful as a pre-processor in applications where directional operators are involved. Rotation is a special case of affine transformation.

The rotation operator performs a transformation of the form:

$$x_2 = cos(\theta) * (x_1 - x_0) - sin(\theta) * (y_1 - y_0) + x_0$$

$$y_2 = sin(\theta) * (x_1 - x_0) + cos(\theta) * (y_1 - y_0) + y_0$$

Four different rotations were applied to every training example with values of **Θ** as

- 45°
- 135°
- 225°
- 315°

Sample images after augmentation have been shown on the next page.

Figure 2 Average Filter

Figure 3 Left Top Cropped

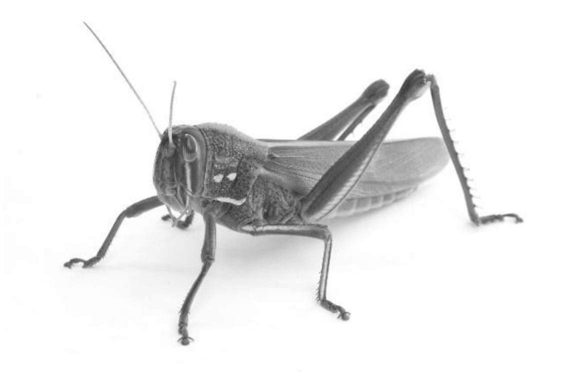


Figure 4 RGB to Grey

Figure 5 Gaussian Filter

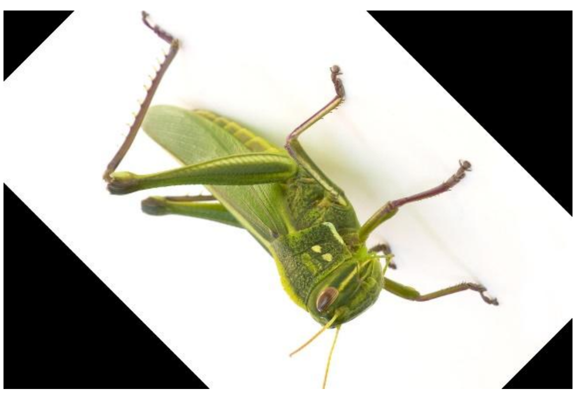


Figure 6 Rotation

Figure 7 Vertical Flip

3.4 Training

3.4.1 Insects identification

After the data acquisition phase, various neural networks were trained in order to get an optimal result for the problem. Initially, a 4 layered neural network was trained from scratch (without transfer learning) with different hyper parameters.

The architecture for the 4 layered neural network is shown below.

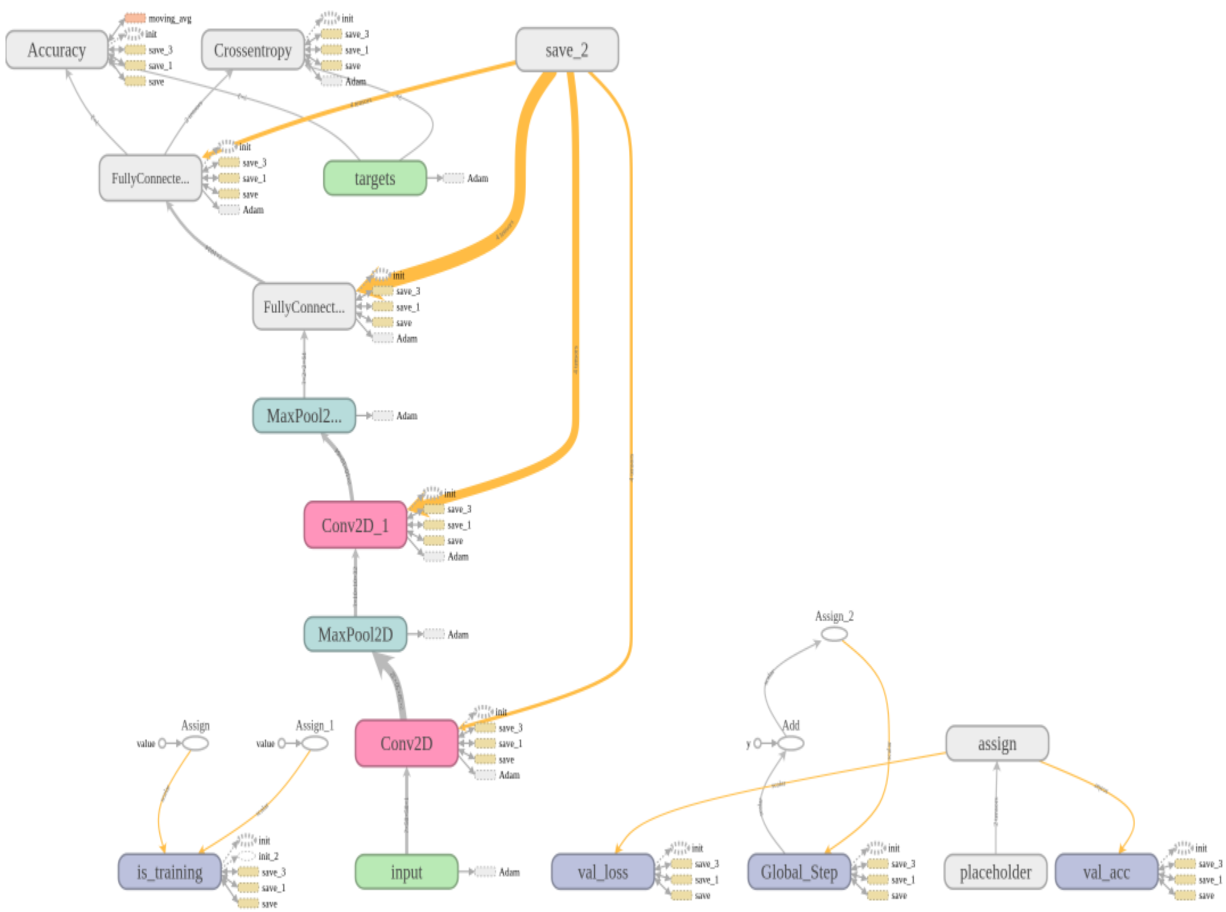


Figure 8 Network architecture for the 4 layered ConvNet

The neural network with above shown network architecture was trained on different hyper parameter choices on a system without GPU support and 16GB RAM. Even after training the model for more than 7 hours, the results were not satisfying.

The reasons behind the poor performance were:

1) Inadequate amount of data for training (The training examples were not enough for insects).
2) Basic network architecture.
3) More data preprocessing was needed.

In order to improve the performance of the ConvNet by using the existing training dataset transfer learning was implemented.

Transfer learning

Make use of the knowledge gained while solving one problem and applying it to a different but related problem is Transfer Learning. When the network is trained on a large dataset (ImageNet), all the parameters of the neural network are trained and therefore the model is learned. Example of one such model is mobilenets. The two major Transfer Learning scenarios look as follows:

ConvNet as fixed feature extractor. Here a ConvNet is pre trained on ImageNet [5], then the last fully connected layer (this layers outputs are the 1000 class scores for a different task like ImageNet) is removed, the rest of the ConvNet is treated as a fixed feature extractor for the new dataset. In an AlexNet, this would compute a 4096-D vector for every image that contains the activations of the hidden layer immediately before the classifier. These features are called as the CNN codes. It is important for performance that these codes are ReLUd (i.e. thresholded at zero) if they were also thresholded during the training of the ConvNet on ImageNet (as is usually the case). Once the 4096-D codes for all images are extracted, a linear classifier (e.g. Linear SVM or Softmax classifier) is trained for the new dataset.

Fine-tuning the ConvNet. The second strategy is to not only replace and retrain the classifier on top of the ConvNet on the new dataset, but to also fine-tune the weights of the pretrained network by continuing the backpropagation [30]. It is possible to fine-tune all the layers of the

ConvNet, or it’s possible to keep some of the earlier layers fixed (due to overfitting concerns) and only fine-tune some higher-level portion of the network. This is motivated by the observation that the earlier features of a ConvNet contain more generic features (e.g. edge detectors or color blob detectors) that should be useful to many tasks, but later layers of the ConvNet become progressively more specific to the details of the classes contained in the original dataset. In case of ImageNet for example, which contains many dog breeds, a significant portion of the representational power of the ConvNet may be devoted to features that are specific to differentiating between dog breeds.

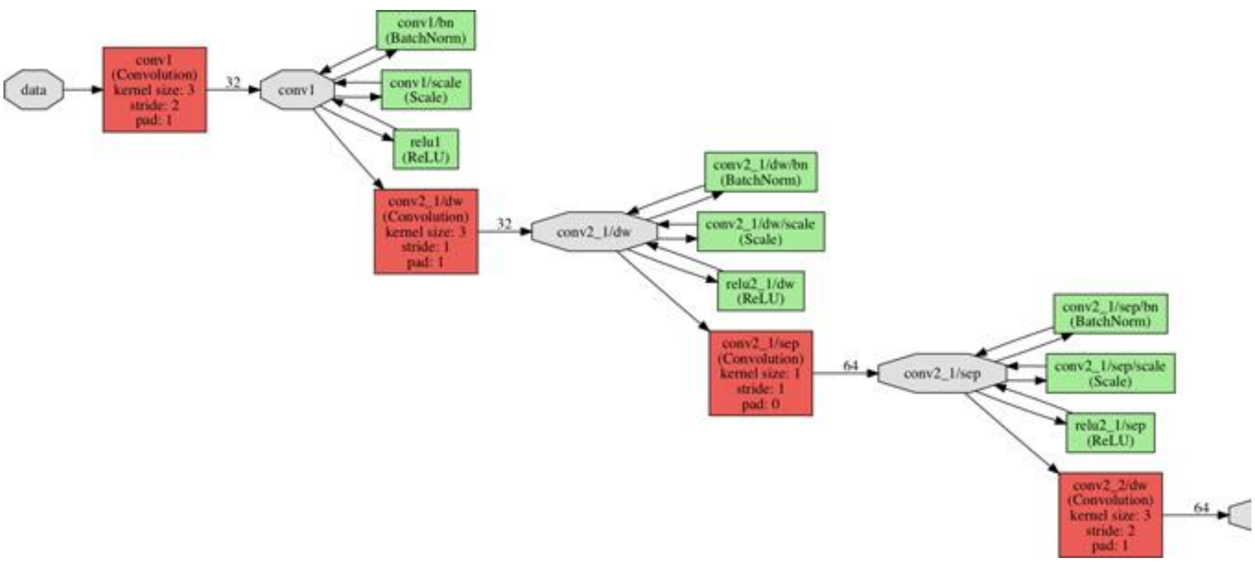


Figure 9 General Architecture of MobileNet 1.0, Trained for Object Classification

Pre-trained Models

Since modern ConvNets take 2-3 weeks to train across multiple GPUs on ImageNet, it is common to see people release their final ConvNet checkpoints for the benefit of others who can use the networks for fine-tuning. For example, the Caffe library has a Model Zoo where people share their network weights.

Since inadequate number of training examples was one of the reasons behind the poor performance of the previous architecture, use of pretrained models was the best solution for this problem because pretrained models don't need to learn the weights from scratch and hence less number of training examples can also provide a satisfiable performance after training.

Some of the pretrained models available out there are:

- Oxford VGG Model
- Google Inception Model
- Mobilenet
- Microsoft ResNet Model
- Google’s word2vec Model
- Stanford’s GloVe Model
- Caffe Model Zoo

The convnet performed best on inceptionV3 with an overall validation accuracy of 91% followed by mobilenet with validation accuracy of 89%. The size of the protobuf file generated by inceptionV3 was 85 Mb whereas for mobilenet0.5 it was 5.4 Mb. Because of the fact that the application was making network calls for the identification, mobilenet was finalised as the pre-trained model to be used in the system in order to reduce the latency during the network calls.

3.4.2 Disease Identification

The number of training examples for classification of plant diseases quite large. Therefore even on the basic 4 layered convnet the validation accuracy was 89% which later shooted up to 96% after implementing transfer learning. But the performance on the real time photos is quite less competent as compared to the photos in the testing set since the training dataset lacks the factor of diversity.

Design and Test Steps

Step: 1) Create Tensorflow session

Step: 2) Create CNN of 15 layers from mobilenet 0.5 weights

Step: 3) Create functions required for Image manipulations viz. loadImage,saveImage etc.. Step: 4) Pass the tensor obtained from the image through CNN

Step: 6) Calculate the cross entropy

Step: 7) Backpropagate loss and minimize it

3.4.3 Application Programming Interface

In order to make the trained model remotely accessible to the Android application, an API is developed which is hosted on the cloud platform called Heroku. The API is written in Flask which provides services such as:

- Identifying the insect from the photo and returning appropriate results in response.
- Identifying the disease from the photo and returning appropriate results in response.
- Saving the labelled images into the database as a crowdsourcing initiative.

The API uses Mongo database for storing all the information related to the insects/diseases and saving the images. The Mongo database for the system is remotely hosted on MLab in an amazon web service instance (In sandbox mode as of now).

Heroku

Heroku is a cloud platform as a service (PaaS) supporting several programming languages that is used as a web application deployment model. Heroku, one of the first cloud platforms, has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go. For this reason, Heroku is said to be a polyglot platform as it lets the developer build, run and scale applications in a similar manner across all the languages.

Applications that are run on Heroku typically have a unique domain (typically "applicationname.herokuapp.com" | “insect-classifier-api.herokuapp.com” in our case) used to route HTTP requests to the correct dyno. Each of the application containers,[3] or dynos,[4] are spread across a "dyno grid" which consists of several servers. Heroku's Git server handles application repository pushes from permitted users.

All Heroku services are hosted on Amazon's EC2 cloud-computing platform.

Flask

Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine. It is BSD licensed.

The latest stable version of Flask is 1.0 as of April 2018.[7] Applications that use the Flask framework include Pinterest, LinkedIn, and the community web page for Flask itself.

Flask is called a micro framework because it does not require particular tools or libraries.[8] It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, and upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more regularly than the core Flask program.

Tensorflow

TensorFlow is an open-source software library for dataflow programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks.[9] It is used for both research and production at Google, often replacing its closed-source predecessor, DistBelief.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open source license on November 9, 2015.

TensorFlow is Google Brain's second generation system. Version 1.0.0 was released on February 11, 2017. While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units).[10] TensorFlow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS.

TensorFlow computations are expressed as stateful dataflow graphs. The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays. These arrays are referred to as "tensors".

TFlearn

TFlearn is a modular and transparent deep learning library built on top of Tensorflow. It was designed to provide a higher-level API to TensorFlow in order to facilitate and speed-up experimentations, while remaining fully transparent and compatible with it.

The high-level API currently supports most of recent deep learning models, such as Convolutions, LSTM, BiRNN, BatchNorm, PReLU, Residual networks, Generative networks... In the future, TFLearn is also intended to stay up-to-date with latest deep learning techniques.

In this project, the initial 4 layered network architectures were built using TFlearn. Whereas In the later development phases the network architecture was built using native tensorflow support.

mLab

mLab is a fully managed cloud database service that hosts MongoDB databases. mLab runs on cloud providers Amazon, Google, and Microsoft Azure, and has partnered with platform-as-a-service providers. MongoDB Inc. provides a fully managed highly available MongoDB-as-a-Service Add-On offering on the Microsoft Azure store. The offering is delivered in collaboration with Microsoft and mLab.

User Endpoints

Base URL: https://insect-classifier-api.herokuapp.com/

Method	Endpoint	Action
PUT	/api/v2/identify/disease	Identify disease and get results
PUT	/api/v2/identify/insect	Identify insect and get results
PUT	/api/v1/save?label=IMAGE_LABEL	Save image with specified label
GET	/api/v2/identify/insect?image_name=IMAGE_URL	Identify insect from image URL
GET	/api/v2/identify/disease?image_name=IMAGE_URL	Identify disease from image URL

Table 1 User Endpoints

3.4.4 JSON response structure:

The JSON response for the request consists of the top 3 predictions in 3 different languages. The individual object consists of the keys such as

- Common name
- Distribution
- Lifespan
- Remedies
- Scientific Name

The sample structure of the JSON response is shown below

```
{
    “Result”:[
        {
            “confidence”:””,
            “english”:{
            “com_name”:””,
            “distribution”:””,
            “lifespan”:””,
            “remedies”:””,
            “sci_name”:””
            },
            “Hindi”:{[+]},
            “Marathi”:{[+]}
```

}   ]   }

3.5 Android Application Development

3.5.1 Android Application User Interface (Quark)

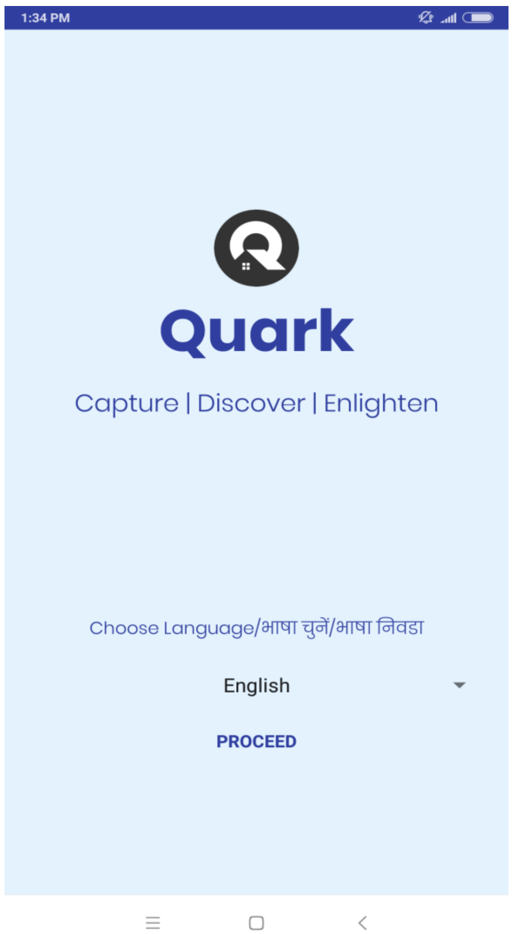


Figure 10 Landing Page

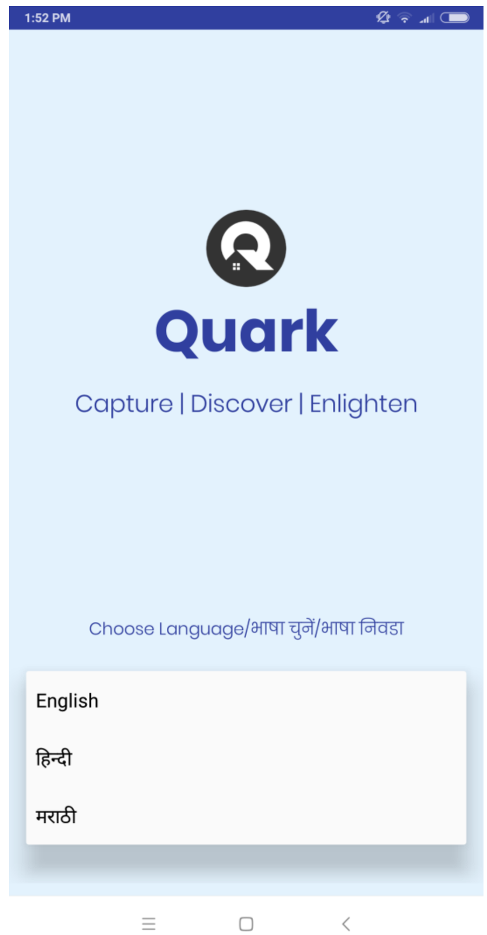


Figure 11 Language Selection

The above figure shows the view of the user interface. The very first feature provided to make the application user friendly is the option of "Choose Language" As the application is to be used by farmers it is necessary to add regional language so as to make the interface user friendly. Choosing the language from this option will change the complete interface to the selected language.

Proceeding further gives the user three options i.e. detect insect, detect disease and add information. The following figure shows the view of the user interface.

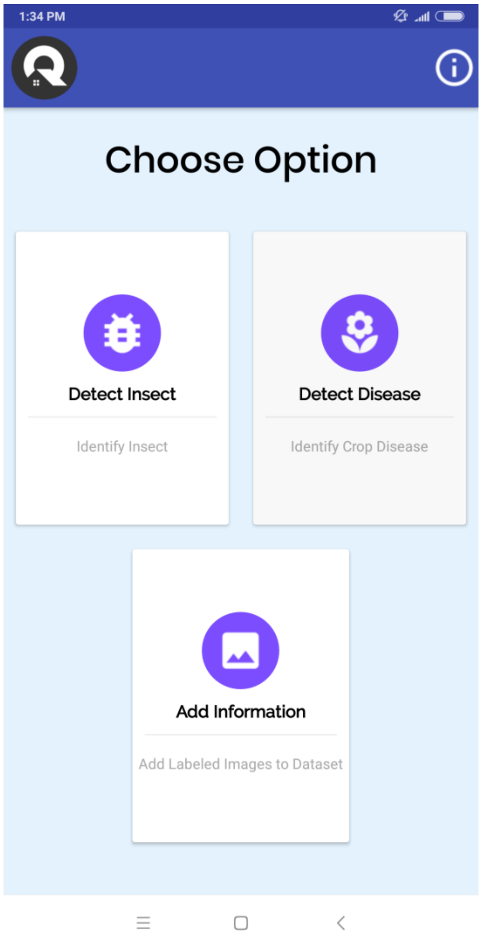


Figure 12 User Choice Page

The third tab of add information is provided to farmers for adding information about new insect or diseases detected. Clicking this tab will open the camera giving two options to click the image or to select the image from the gallery. After selection of the image the user can submit the image with the known information. The following figures show the view of the interface providing these options.

Figure 13 Image Capture Screen

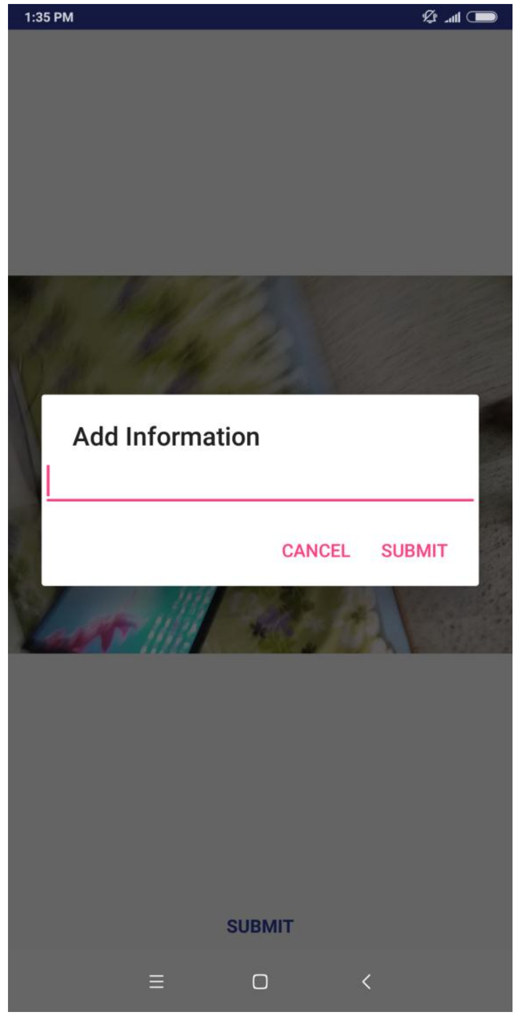


Figure 14 Add Label Information

Similarly further options of detect insect and detect disease also open the camera and from here the user can submit the input. After submitting the input if the input is appropriate the user gets the feedback which includes all the important information about the insect or disease.

Following figure show the sample of feedback for the image of Grasshopper

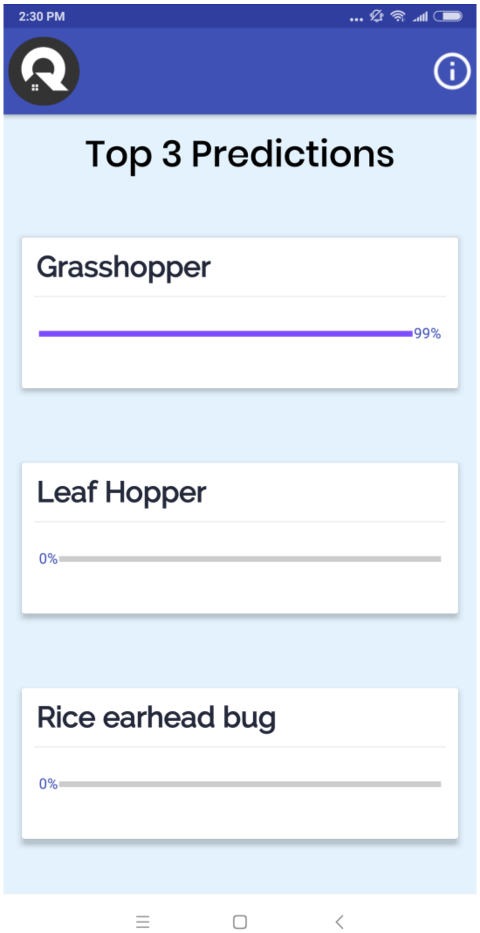


Figure 15 Top 3 Predictions as identified by Model

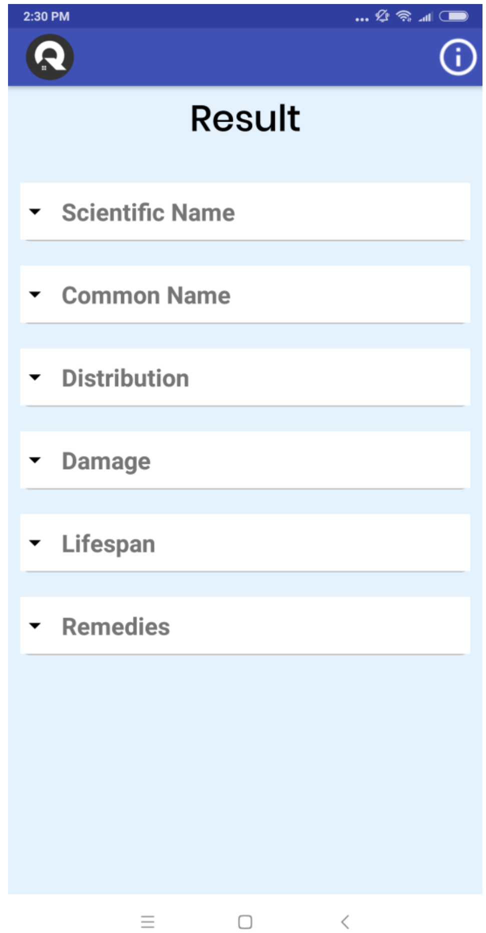


Figure 16 Result Page

The Result page displays useful information related to the insect or crop disease identified, like Common Name, Scientific Name, Damage, Distribution, Lifespan, Remedies.

3.5.2 Restful API

A Restful API is an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data.

A Restful API - also referred to as a Restful web service -- is based on representational state transfer (REST) technology, an architectural style and approach to communications often used in web services development.

REST technology is generally preferred to the more robust Simple Object Access Protocol (SOAP) technology because REST leverages less bandwidth, making it more suitable for internet usage. An API for a website is code that allows two software programs to communicate with each another. The API spells out the proper way for a developer to write a program requesting services from an operating system or other application.

The REST used by browsers can be thought of as the language of the internet. With cloud use on the rise, APIs are emerging to expose web services. REST is a logical choice for building APIs that allow users to connect and interact with cloud services. Restful APIs are used by such sites as Amazon, Google, LinkedIn and Twitter.

How Restful APIs work?

A Restful API breaks down a transaction to create a series of small modules. Each module addresses a particular underlying part of the transaction. This modularity provides developers with a lot of flexibility, but it can be challenging for developers to design from scratch. Currently, the models provided by Amazon Simple Storage Service, Cloud Data Management Interface and Open Stack Swift are the most popular.

Use of Camera API in Android Application (Quark):

User interface of the android application makes use restful API for uploading image from mobile to server and to receive the JSON object from the server.

3.5.3 Camera API

The Android framework includes support for various cameras and camera features available on devices, allowing you to capture pictures and videos in your applications.

Building a camera app:

Some developers may require a camera user interface that is customized to the look of their application or provides special features. Writing your own picture-taking code can provide a more compelling experience for your users.

The general steps for creating a custom camera interface for your application are as follows:

- Detect and Access Camera - Create code to check for the existence of cameras and request access.
- Create a Preview Class - Create a camera preview class that extends SurfaceView and implements the SurfaceHolder interface. This class previews the live images from the camera.
- Build a Preview Layout - Once you have the camera preview class, create a view layout that incorporates the preview and the user interface controls you want.
- Setup Listeners for Capture - Connect listeners for your interface controls to start image or video capture in response to user actions, such as pressing a button.
- Capture and Save Files - Setup the code for capturing pictures or videos and saving the output.
- Release the Camera - After using the camera, your application must properly release it for use by other applications.

Use of Camera API in Android Application (Quark)

We designed an Activity which is used for capturing images using camera API.

It consists of 2 options one for capturing and other to choose image from gallery.

3.5.4 JSON

JSON: JavaScript Object Notation. JSON is a syntax for storing and exchanging data. JSON is text, written with JavaScript object notation.

Exchanging Data

When exchanging data between a browser and a server, the data can only be text. JSON is text, and we can convert any JavaScript object into JSON, and send JSON to the server. We can also convert any JSON received from the server into JavaScript objects. This way it is possible to work with the data as JavaScript objects, with no complicated parsing and translations.

Why use JSON?

Since the JSON format is text only, it can easily be sent to and from a server, and used as a data format by any programming language. JavaScript has a built in function to convert a string, written in JSON format, into native JavaScript objects:

JSON.parse()

So, if the data received from a server, in JSON format, it can be used like any other JavaScript object.

Use of JSON object in Android Application (Quark)

The mobile app receives the JSON object as a response from server which consists of all the information related to the predictions obtained from the module. The JSON object is then unwrapped and displays the information to the user.

3.5.5 System Architecture

Shown below are diagrams that explain the flow and working of the android application.

The architecture of the application is as follows:

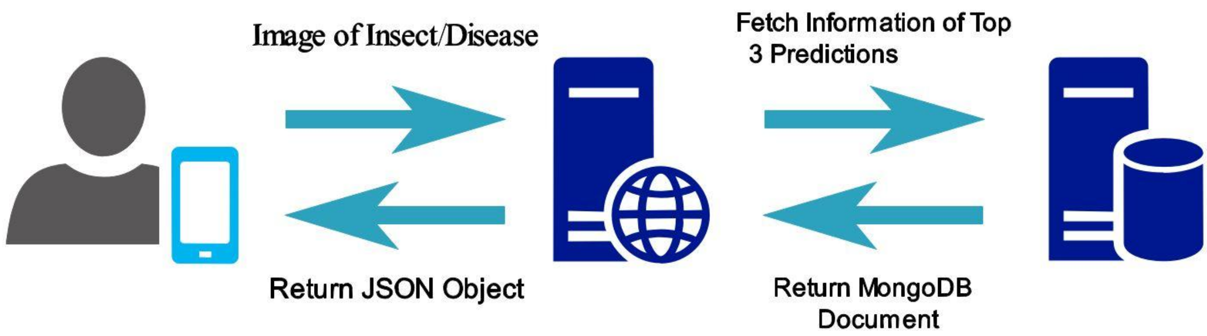


Figure 17 Application Architecture

The Use Case Diagram of the application is shown below:

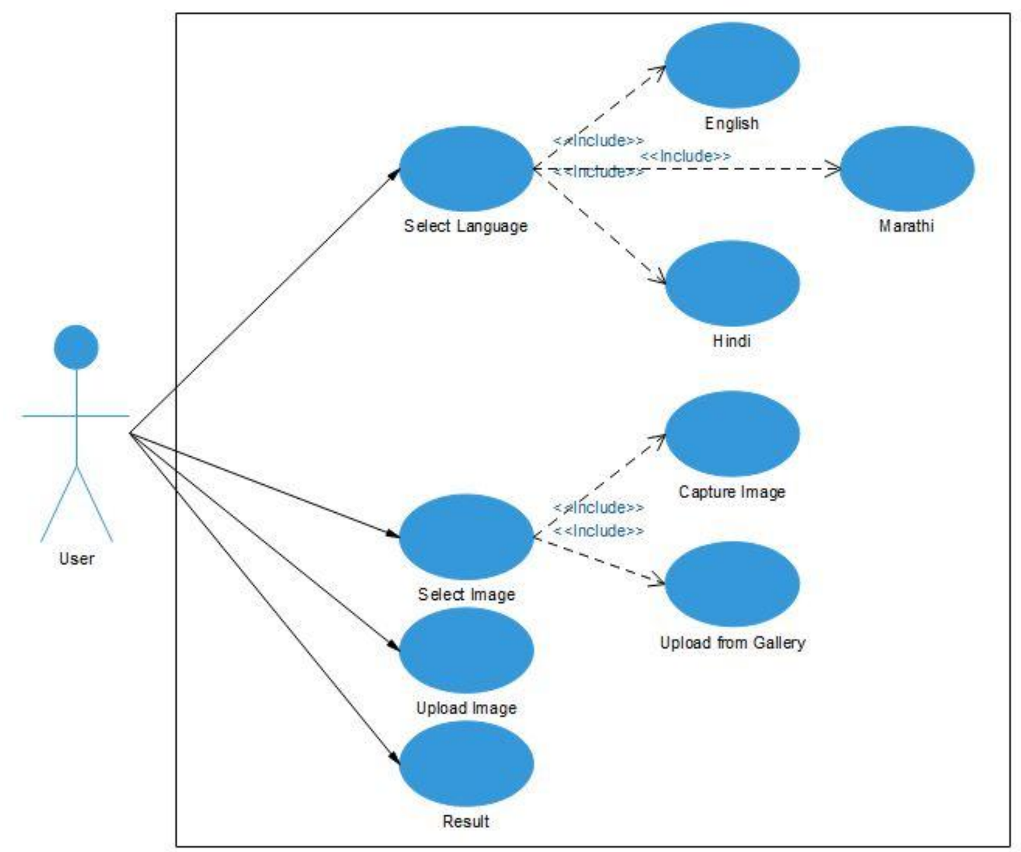


Figure 18 Use Case Diagram

The flow of control in the application is depicted in the flow chart below:

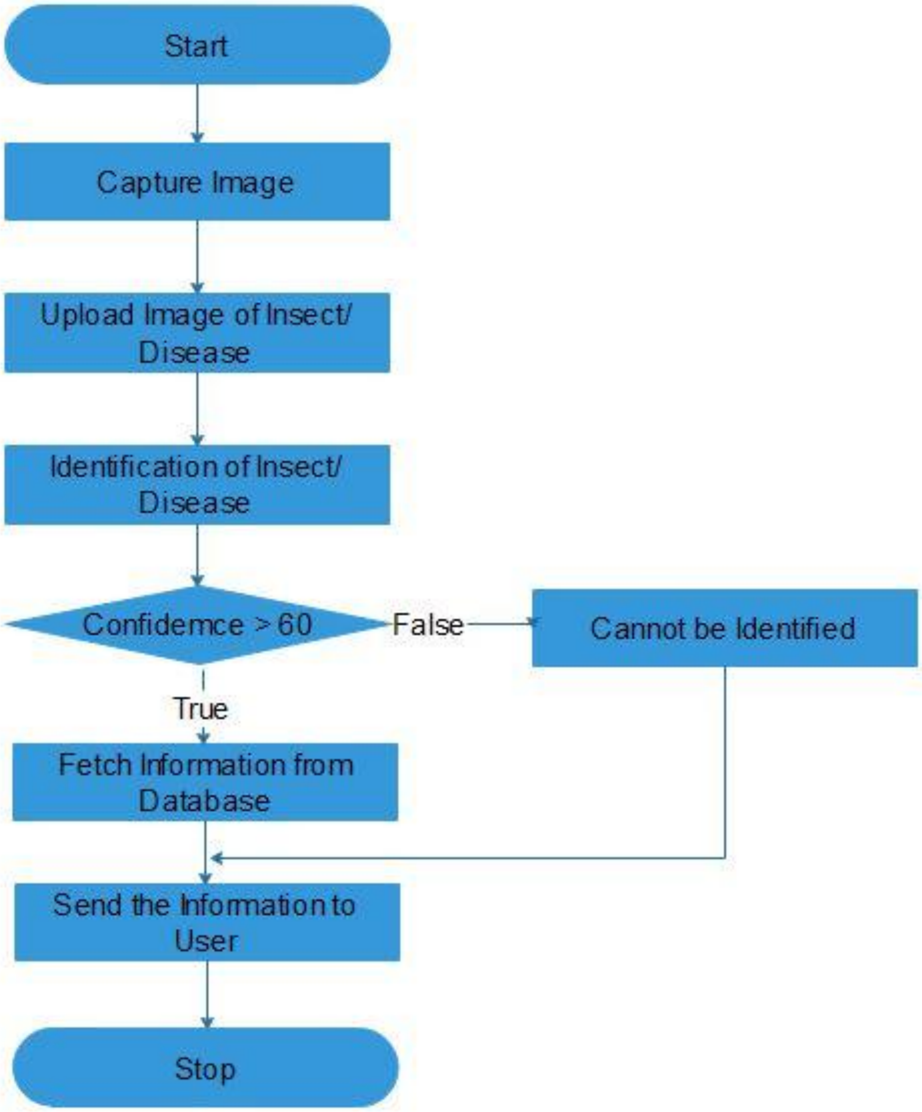


Figure 19 Flow of Control in the Application

3.5.6 App Development Repositories

Android accordion view

Using the android accordion component developer set the heading and the UI elements in the body right from the layout xml or from the class. This component also provide various other UI features like animation to be controlled by the developer from the xml itself.

Android Number Progress Bar

Android Progress Bar is a graphical view indicator that shows some progress. Android progress bar displays a bar representing the completing of the task. Progress bar in android is useful since it gives the user an idea of time to finish its task.

Using a Progress Bar is a good user experience practice since it displays the status of progress of the given task (such as downloading an image) to the user.

Some important attributes used to describe a ProgressBar are given below.

- android:max : We can set the maximum value of the ProgressBar using this attribute. By default the progress bar maximum value is 100
- android:indeterminate : A boolean value is set depending on whether the time is determinate or not. Setting this attribute to false would show the actual progress. Else if it’s set to true a cyclic animation is displayed to show that progress is happening
- android:minHeight : It’s used to set the height of the ProgressBar
- android:minWidth : It’s used to set the width of the ProgressBar
- android:progress : It’s used to set the number by which the progress bar value will be incremented
- style: By default the progress bar will be displayed as a spinning wheel. If we want it to be displayed as a horizontal bar, we need to set the attribute as : style=”?android:attr/progressBarStyleHorizontal”

Android Loading Spinner

Progress of a task in android can be shown through loading progress bar. The progress bar comes in two shapes. Loading bar and Loading Spinner. In this chapter we will discuss spinner.

Spinner is used to display progress of those tasks whose total time of completion is unknown

Methods used in Android Loading Spinner

- isIndeterminate()

Indicate whether this progress bar is in indeterminate mode

- postInvalidate()

Cause an invalidate to happen on a subsequent cycle through the event loop

- setIndeterminate(boolean indeterminate)

Change the indeterminate mode for this progress bar

- incrementSecondaryProgressBy(int diff)

Increase the progress bar's secondary progress by the specified amount

- getProgressDrawable()

Get the drawable used to draw the progress bar in progress mode

Android Material Design

Material design is a comprehensive guide for visual, motion, and interaction design across platforms and devices. To use material design in your Android apps, follow the guidelines defined in the material design specification and use the new components and styles available in the material design support library. This page provides an overview of the patterns and APIs you should use. Android provides the following features to help you build material design apps:

- A material design app theme to style all your UI widgets
- Widgets for complex views such as lists and cards
- New APIs for custom shadows and animations

3.6 Database Schema

In order to return appropriate information along with the predicted results a database was created which contained all the textual data in three supported languages English, Hindi, Marathi.

In order to store images in MongoDB we wrote a script to encode it into a string format for easy storage and retrieval. If images exceed the threshold of 16MB GridFS provision is available with MongoDB.

Since our requirement was to store data which varied in length for each unique document, traditional SQL database would be tedious and would not suffice. Also, since the crowdsourcing module of the system requires the user to upload an image to the database, traditional SQL was not the right technology to choose. Instead we chose NoSQL to be implemented through MongoDB which allowed enough flexibility and freedom to store semi-structured data as well as support a variety of formats.

Document Store Databases stores the data in the form of documents. Document Store databases are schema less, so they are much more flexible compared to the records in relational databases. Format of documents are PDF, XML, JSON etc. Each document in document store databases is addressed by unique key for representation of that document. It is used in such applications where data need to be store as documents that having some special characteristics.

MongoDB is proposed by 10gen Company in order to handle growing data storage needs. MongoDB is open source NoSQL document store database which is written in C++. MongoDB uses JavaScript as its query language. Data is stored in MongoDB in the form of collections. Each collection consists of documents. MongoDB stores the documents in BSON format which is binary form of JSON. BSON supports different data types such as integer, float, string, Boolean, date etc. MongoDB is schema-less as it is having document structure. To distribute the collections across multiple nodes MongoDB offers sharding technique. MongoDB automatically redistribute the data across the nodes, thus the load is balanced and equally distributed over the nodes. MongoDB allows Master-Slave replication technique. Here the Slaves are the nodes that contain the copies of Master nodes and it is used for backup process and read operations.

Sample document of the Insect database is shown below:

{

"_id": {

"$oid": "5ab3e2a8734d1d1aa1538a20"

},

"sci_name": "carpenter ant",

"english": {

"sci_name": "Camponotus",

"com_name": "carpenter ant",

"distribution": "Camponotus vicinus (Mayr) is a common ant species found in the Pacific Northwest. It is an important predator of many forest insect pests, a potential biological control agent, and is also a serious structural pest.",

"damage": "Contrary to popular belief, carpenter ants do not actually eat the wood. Rather, they hollow it out in order to nest inside, which may result in structural damage. Also unlike termites, carpenter ants generally take years to cause significant damage.",

"lifespan": "It takes three to six years to establish a large and stable colony. The life cycle of a carpenter ant is estimated to be 6 to 12 weeks from egg to adult. Cold weather can stretch the development time of carpenter ants up to 10 months.",

"remedies": "Mix 1 part boric acid with 10 parts sugar water, add this mixture to the food you want to use as bait, and set it out along any carpenter ant trails or spots you think foraging workers frequent. The sugar water in the mixture will draw the workers in, and the boric acid will kill them \u2013 and their nest."

},

"hindi": {},

"marathi": {}}

4. Results and Performance Analysis:

4.1 Evaluation Measures

Confusion matrix

A confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa). The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another).

It is a special kind of contingency table, with two dimensions ("actual" and "predicted"), and identical sets of "classes" in both dimensions (each combination of dimension and class is a variable in the contingency table).

Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. It can be described as a measure of a classifiers exactness. A low precision can also indicate a large number of False Positives.

Precision = TP/TP+FP

Recall

Recall is the number of True Positives divided by the number of True Positives and the number of False Negatives. Put another way it is the number of positive predictions divided by the number of positive class values in the test data. It is also called Sensitivity or the True Positive Rate.

Recall = TP/TP+FN

Accuracy

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. Accuracy is substantially great measure but only in case of symmetric datasets where values of false positive and false negatives are almost same.

Accuracy = TP+TN/TP+FP+FN+TN

F1-Score

F1 score - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it’s better to look at both Precision and Recall.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

4.2 Experimental Results

4.2.1 Insect classification model

Below shown are the experimental results for the classifier trained using transfer learning. Overall precision and recall for the model is 87%, f1 score is 86% which was calculated over the support of 975 testing examples.

The labels for

Confusion Matrix:

```
[[33,  0, 18,  0,  2,  5,  2,  0,  0,  0,  1,  0,  2,  1,  1],
 [ 0, 63,  0,  0,  0,  0,  0,  1,  0,  0,  0,  0,  1,  0,  0],
 [14,  0, 50,  0,  0,  0,  1,  0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0, 64,  1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0, 58,  0,  2,  1,  0,  0,  0,  0,  3,  0,  1],
 [ 1,  1,  1,  0,  7, 42,  3,  3,  0,  0,  0,  0,  7,  0,  0],
 [ 0,  0,  0,  0,  0,  0, 65,  0,  0,  0,  0,  0,  0,  0,  0],
 [ 0,  1,  0,  0,  0,  0,  0, 57,  1,  0,  4,  0,  1,  1,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0, 65,  0,  0,  0,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  0,  0,  0,  0, 61,  0,  0,  4,  0,  0],
 [ 0,  0,  0,  0,  1,  0,  0,  0,  0,  0, 61,  0,  3,  0,  0],
 [ 0,  0,  2,  0,  0,  0,  0,  0,  0,  0,  0, 61,  2,  0,  0],
 [ 0,  2,  2,  0,  3,  3,  0,  1,  0,  1,  1,  4, 47,  0,  1],
 [ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, 65,  0],
 [ 0,  0,  0,  1,  0,  0,  1,  4,  0,  0,  0,  1,  1,  4, 53]]
```

Figure 20 Confusion Matrix for the insect classifier

Classification report

	precision	recall	f1-score	support
ArmyWorm	0.69	0.51	0.58	65
CarpenterAnt	0.94	0.97	0.95	65
CornEarworm	0.68	0.77	0.72	65
Earwig	0.98	0.98	0.98	65
Grasshopper	0.81	0.89	0.85	65
Leafhopper	0.84	0.65	0.73	65
Leptocorisaacuta	0.88	1.00	0.94	65
Muscidae	0.85	0.88	0.86	65
Mutillidae	0.98	1.00	0.99	65
OilBeetle	0.98	0.94	0.96	65
Oryctesrhinoceros	0.91	0.94	0.92	65
Weevil	0.92	0.94	0.93	65
Woodborer	0.66	0.72	0.69	65
WoollyBearMoth	0.92	1.00	0.96	65
WorkerBee	0.95	0.82	0.88	65
avg / total	0.87	0.87	0.86	975

Table 2 Classification Report

Tensorflow plots

The figures below show the plots of the progress of the model's accuracy and cross entropy as it trains. Two lines are shown. The orange line shows the accuracy of the model on the training data. While the blue line shows the accuracy on the test set (which was not used for training). This is a much better measure of the true performance of the network. If the training accuracy continues to rise while the validation accuracy decreases then the model is said to be "overfitting". Overfitting is when the model begins to memorize the training set instead of understanding general patterns in the data.

As it can be seen that after the training the model completely, the validation accuracy is 88.4% which might change by ∓2 every time since there is randomness in the training process. This number value indicates the percentage of the images in the test set that are given the correct label after the model is fully trained.

The first figure shows accuracy (x-axis) as a function of training progress (y-axis):

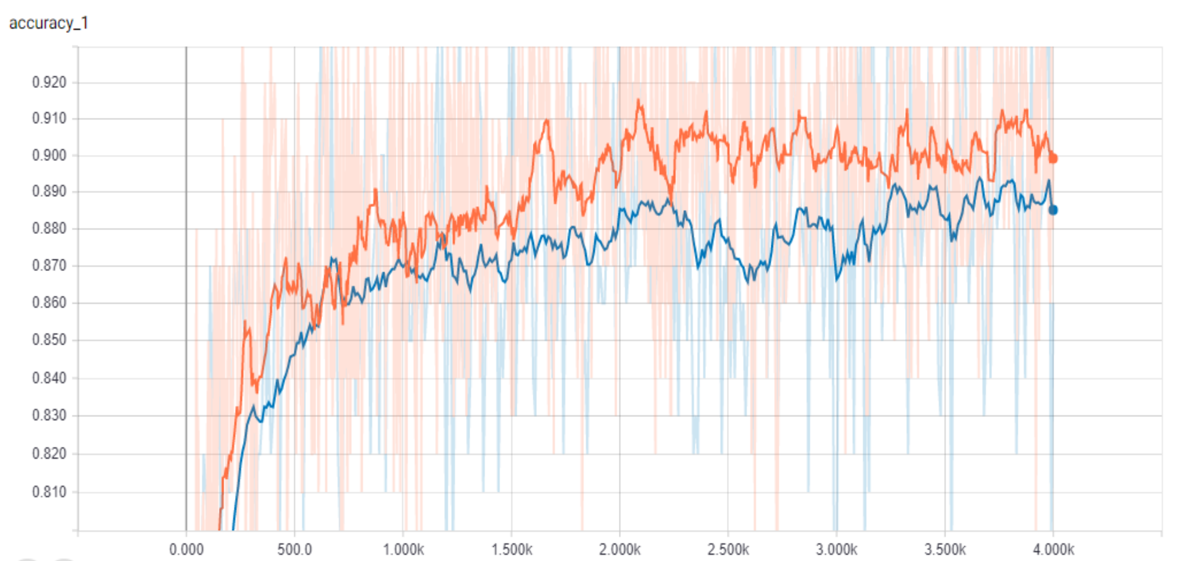
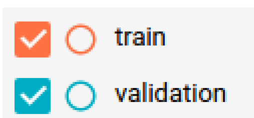


Figure 21 Accuracy vs. Time (In seconds) plot for training phase.



Below shown is the plot for the cross entropy vs time (in seconds) for the training phase. The cross entropy used for this model was Softmax. The value of the cross entropy decreases over the epochs until convergence.

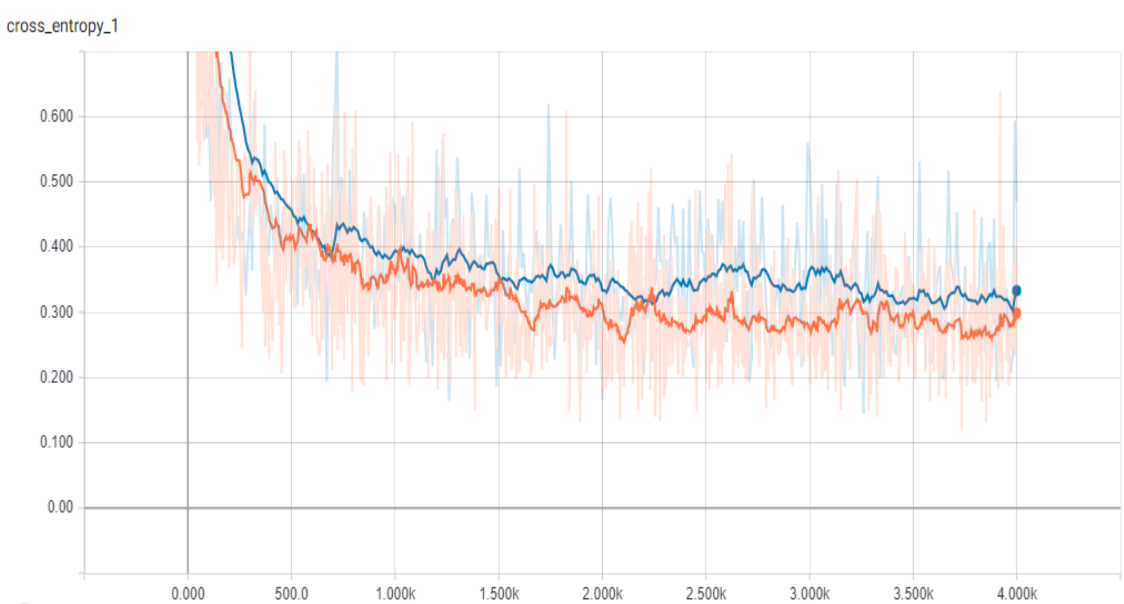


Figure 22 Cross entropy vs. Time (In seconds) plot for training phase.

4.2.2 Disease classification model

Confusion Matrix

Shown below is the confusion matrix for the disease classifier. The rows represent the actual classes whereas the columns represent the predicted classes. The labels for the matrix are in the sequence as follows. The precision for the model 0.98, recall and f1 score is 0.97 which is calculated over the support of 1110 testing examples.

1. Apple___Apple_scab
2. Apple___Black_rot
3. Apple___Cedar_apple_rust
4. Apple___healthy
5. Blueberry___healthy
6. Cherry_(including_sour)___Powdery_mildew
7. Cherry_(including_sour)___healthy
8. Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot
9. Corn_(maize)___Common_rust_
10. Corn_(maize)___Northern_Leaf_Blight
11. Corn_(maize)___healthy
12. Grape___Black_rot
13. Grape___Esca_(Black_Measles)
14. Grape___Leaf_blight_(Isariopsis_Leaf_Spot)
15. Grape___healthy
16. Orange___Haunglongbing_(Citrus_greening)
17. Peach___Bacterial_spot
18. Peach___healthy
19. Pepper,_bell___Bacterial_spot
20. Pepper,_bell___healthy
21. Potato___Early_blight
22. Potato___Late_blight
23. Potato___healthy
24. Raspberry___healthy
25. Soybean___healthy
26. Squash___Powdery_mildew
27. Strawberry___Leaf_scorch
28. Strawberry___healthy
29. Tomato___Bacterial_spot
30. Tomato___Early_blight

31. Tomato___Late_blight
32. Tomato___Leaf_Mold
33. Tomato___Septoria_leaf_spot
34. Tomato___Spider_mites Two-spotted_spider_mite
35. Tomato___Target_Spot
36. Tomato___Tomato_Yellow_Leaf_Curl_Virus
37. Tomato___Tomato_mosaic_virus
38. Tomato___healthy

```
[[30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 28  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 1  0  0 28  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0 29  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 23  0  5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 29  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0 29  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0 28  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0 27  2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 27  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 29  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 28  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 27  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 28  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1 28  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 28  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 29  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 28  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 27  0  0  0  1  0  1  1  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 26  1  1  1  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 29  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 26  1  1  0  0  1  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0 28  0  0  0  0  1]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 26  1  0  0  1]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0 29  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 29  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 29]]
```

Figure 23 Confusion Matrix for disease classifier

Classification report

	precision	recall	f1-score	support
Apple___Apple_scab	0.97	1.00	0.98	30
Apple___Black_rot	1.00	1.00	1.00	28
Apple___Cedar_apple_rust	1.00	1.00	1.00	30
Apple___healthy	1.00	0.93	0.97	30
Blueberry___healthy	1.00	1.00	1.00	29
Cherry_(including_sour)___Powdery_mildew	1.00	1.00	1.00	30
Cherry_(including_sour)___healthy	1.00	1.00	1.00	30
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot	1.00	0.82	0.90	28
Corn_(maize)___Common_rust_	1.00	1.00	1.00	29
Corn_(maize)___Northern_Leaf_Blight	0.85	1.00	0.92	29
Corn_(maize)___healthy	1.00	1.00	1.00	28
Grape___Black_rot	1.00	0.93	0.96	29
Grape___Esca_(Black_Measles)	0.94	1.00	0.97	30
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	0.97	1.00	0.98	30
Grape___healthy	1.00	1.00	1.00	30
Orange___Haunglongbing_(Citrus_greening)	1.00	0.96	0.98	28
Peach___Bacterial_spot	1.00	0.97	0.98	30
Peach___healthy	0.94	1.00	0.97	30
Pepper,_bell___Bacterial_spot	1.00	1.00	1.00	30
Pepper,_bell___healthy	1.00	1.00	1.00	28
Potato___Early_blight	1.00	1.00	1.00	27
Potato___Late_blight	0.97	0.97	0.97	29
Potato___healthy	1.00	0.97	0.98	29
Raspberry___healthy	1.00	1.00	1.00	28
Soybean___healthy	1.00	1.00	1.00	30
Squash___Powdery_mildew	1.00	1.00	1.00	29
Strawberry___Leaf_scorch	1.00	1.00	1.00	30
Strawberry___healthy	1.00	1.00	1.00	28
Tomato___Bacterial_spot	1.00	0.90	0.95	30
Tomato___Early_blight	0.93	0.87	0.90	30
Tomato___Late_blight	0.91	1.00	0.95	29
Tomato___Leaf_Mold	0.96	0.90	0.93	29
Tomato___Septoria_leaf_spot	0.90	0.93	0.92	30
Tomato___Spider_mites Two-spotted_spider_mite	0.96	0.93	0.95	28
Tomato___Target_Spot	0.94	0.97	0.95	30
Tomato___Tomato_Yellow_Leaf_Curl_Virus	0.97	1.00	0.98	29
Tomato___Tomato_mosaic_virus	0.97	1.00	0.98	30
Tomato___healthy	0.94	1.00	0.97	29
avg / total	0.98	0.97	0.97	1110

Table 3 Classification Report for Disease Classifier

5. Conclusion

5.1 Conclusion

The proposed technological solution is a convolutional neural network trained using a dataset of insect and crop disease images for the image based classification. The above system is a supervised learning model using deep learning. The system is an Android application at the user's end. The user will capture the image of insect or crop disease growing on the crop, through the application. These 2D images will act as an input to the algorithm. The image will then be processed on a server where it will be classified into one of the classes return a result containing the top three classes predicted. Along with the predicted classes important data related to those classes will be sent to the user like Common Name, Scientific Name, Damage, Distribution, and Preventive measures.

Web scraping and ImageNet was crucial and useful resource for collection of training data and data augmentation techniques on those datasets helped improve the accuracy of both models substantially. The model trained on insect images now has a validation accuracy of 87% while the model trained on crop disease images had a validation accuracy of 96%.

5.2 Future Scope

This project carries immense promise because it is a technological solution that can improve the quality of life for farmers all across the country, boost their crop yield and ensure that time lags do not increase the damage caused.

With the emergence of a data centric world we can foresee that in the coming years images of many more insects and crop diseases will be available in abundance which will help train the convolutional neural network model of this system over a larger variety of classes and widen its scope.

Also the hyper parameters of the existing model can be manipulated and modified to attempt an even further increase of accuracy. Layers can be added to the existing architecture to improve it for the better.

UI and application level improvements are also possible where the crowdsourcing and user experience of the application is concerned.

This technological solution is clearly an idea for the future with immense potential to revolutionise the society provided it is equipped with the right vision.

6. References

[1] R, Ujagir & Byrne, Oonagh. (2009). Insect pests and their management. The Lentil: Botany, Production and uses. 282-305.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'12), F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.), Vol. 1. Curran Associates Inc., USA, 1097-1105.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei (Dept. of Computer Science, Princeton University, USA). ImageNet: A Large-Scale Hierarchical Image Database

[4] Peng, Dunlu & Cao, Lidong & Xu, Wenjie. (2011). Using JSON for Data Exchanging in Web Service Applications. 7.

[5] Suhas Holla, Mahima M Katti."Android Based Mobile Application Development and Its Security "***International Journal of Computer Trends and Technology (IJCTT)***, V3 (3):1077-1080 Issue 2012 .ISSN 2231-2803.www.ijcttjournal.org. Published by Seventh Sense Research Group.

[6] Amarpreet Singh Johal and Baljit Singh. Article: Performance Analysis of Web Services for Android based Devices. *International Journal of Computer Applications* 92(11):43-46, April 2014.

[7] G. Selvakumar and B. J. Kaviya, "A survey on RESTful web services composition," *2016 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, 2016, pp. 1-4.

[8] Supriya S. Pore, Swalaya B. Pawar, “Comparative Study of SQL & NoSQL Databases”, International Journal of Advanced Research in Computer Engineering & Technology, ISSN: 2278 – 1323, Volume 4 Issue 5, May 2015, pp: 1747- 1753

[9] Wang, J., & Perez, L. (2017). The effectiveness of data augmentation in image classification using deep learning. Tech. rep., Technical report.