E-Agri Kit: Agricultural Aid Using Deep Learning

K PRASAD[1], B. MURALI[2]
[1]*PG Student, CN, Quba College of Engineering & Technology*
[2]*Associate professor, CN, Quba College of Engineering & Technology*

Abstract*— This project presents an agricultural aid application, developed and designed, to help farmers by utilizing Image Processing, Machine Learning and Deep Learning concepts. Our application provides features such as early detection of plant disease, implemented using various approaches. After evaluation, results showed that Convolutional Neural Network was performing better for plant disease detection with an high accuracy. It further helps the farmer to forecast the weather to decide the right time for agricultural activities like harvesting and plucking. To avoid reoccurrence of disease due to loss in soil minerals, a crop specific fertilizer calculator is incorporated which can calculate the amount of urea, diammonium phosphate and muriate of potash required for a given area.

Index Terms*—Agricultural Aid Application, Plant Disease Detection, Image Processing, Machine Learning, Deep Learning, Convolutional Neural Network (CNN), Weather Forecasting, Fertilizer Calculator, Precision Agriculture Smart Farming

I. INTRODUCTION

According to a study by the Associated Chambers of Commerce and Industry of India, annual crops losses due to pests and diseases amount to Rs.50,000 crore ($500 billion), which is tantamount to a country where at least 200 million go to bed hungry every night [1]. Agriculture being a vital sector has a majority of the rural population in developing countries relying on it. The sector is faced by major challenges like unprecedented pest attack and unforeseen weather conditions affecting their produce leading to major loss of food and effort. Technology plays a vital role in uplifting the livelihoods of the rural populace which can be done by using a simple agro-android application system.

Plant diseases can affect vast produce of crops posing a major menace to food security as well as leading to major losses to farmers. An extensive review of existing research was conducted by us on this domain [5] and in an effort to help farmers overcome this problem, we have designed an android application, Agricultural Aid which utilizes machine learning to provide plant disease detection. This detection is combined with an android application which provides features like weather forecast of up to 7 days, fertilizer calculator and language translation in up to 4 languages which has been implemented and integrated using Android Studio and its APIs. For disease classification, we followed two approaches: Image Processing with Machine Learning and Deep Learning models.

The first approach i.e. Image Processing approach usually includes multi-step preprocessing techniques such as: Filtering, color space conversion, thresholding and finally, contouring to mark out the infected region. These methods can be used with Machine Learning concepts to provide classification of infected regions. However, the accuracy for such methods isn't very high. As an alternative to these steps, "GrabCut" Algorithm can also be used which is an optimized method of foreground extraction to eliminate background noises using minimal user interaction [4]. It has better accuracy in terms of background elimination and can be used for better classification however for the time being this method wasn't used in the application but can be incorporated in the future to improve accuracy. For the second approach i.e. Deep Learning approach, a deep neural architecture is used to train and test on leaf image databases to classify the disease.

II. LITERATURE SURVEY

2.1 INTRODUCTION:

2.1 V. Singh, V. and P. A. K. Misra , "Detection of unhealthy region of plant leaves using Image Processing and Genetic Algorithm," in 2015 International Conference on Advances in Computer Engineering and Applications (ICACEA), Ghaziabad, India, 2015. Agricultural productiveness is that element on which Indian Economy fantastically depends. This is the one of the motives that ailment detection in plant life performs an essential position in agriculture field, as having sickness in plant life are

pretty natural. If suitable care is now not taken in this vicinity then it reasons serious outcomes on flowers and due to which respective product quality, volume or productiveness is affected. Detection of plant ailment via some automated method is really helpful as it reduces a giant work of monitoring in massive farms of crops, and at very early stage itself it detects the signs and symptoms of ailments skill when they show up on plant leaves. This paper provides an algorithm for photo segmentation approach used for computerized detection as nicely as classification of plant leaf ailments and survey on exceptional illnesses classification strategies that can be used for plant leaf sickness detection. Image segmentation, which is an necessary component for disorder detection in plant leaf disease, is executed by using the use of genetic algorithm

2.2 W. Ding and G. Taylor, "Automatic moth detection from trap images for pest management", Computers and Electronics in Agriculture, vol. 123, pp. 17-28, 2016. Available: 10.1016/j.compag.2016.02.003.

We advocate a convolutional neural network-based automated moth detection pipeline. We describe a set of strategies for preprocessing uncooked moth entice images. Our approach indicates promising overall performance on a codling moth dataset from photographs amassed in the field. Our species agnostic technique can be effortlessly tailored to one-of-a-kind pests and/or environments. Monitoring the range of insect pests is a quintessential thing in pheromone-based pest administration systems. In this paper, we suggest an computerized detection pipeline based totally on deep mastering for figuring

III. SYSTEM ANALYSIS

3.1 EXISTING METHOD

Image Processing is a popular first step in the process of plant disease detection and often includes multi-step processes to achieve processed, ROI centric input images for further classification. In this approach, various Image Processing techniques were used on the input image to get a final output image which would mark the infected area and also calculate the percentage of area infected in the leaf. The major advantage observed in this approach was that this approach eliminated the need to extract the leaf and place it on a black background during image capture for the algorithm to work. In real time scenario, the infected leaf would be present amongst a cluster of mixed crops and this approach was able to segregate the infected leaf from the healthy ones in an input image.

DRAWBACKS:

- Accuracy is Low.
- This multistep algorithm is that it is not optimized and no better background elimination results.

PROPOSED METHOD

In the Deep Learning Approach, we decided to take a subset of the Plant Village dataset along with the cotton dataset to train and test the CNN model. The input image is fed into this model, which initially took a portion of the Plant Village and Cotton Dataset, with a training -validation split of 70-30, therefore getting 4200 images for training and 1800 images for validation. A CNN (Convolutional Neural Network) is a deep learning model that takes inputs which are assigned weights depending on various features. CNN is a widely used neural network for image-based datasets.

Our CNN model consists of 4 main convolutional layers with 32, 64, 128, 128 filters consecutively, each followed by a ReLU activation function, max pooling and dropout layer. This set of convolutional layers is followed by a flatten and then a dense layer which is finally followed by the softmax activation function that tells us which class has the maximum probability.

3.3. SYSTEM REQUIREMENTS

Functional requirements for a secure cloud storage service are straightforward:

1.The service should be able to store the user's data;

2.The data should be accessible through any devices connected to the Internet;

3.The service should be capable to synchronize the user's data between multiple devices (notebooks, smart phones, etc.);

4.The service should preserve all historical changes (versioning);

5.Data should be shareable with other users;

6.The service should support SSO; and

7.The service should be interoperable with other cloud storage services, enabling data migration from one CSP to another.

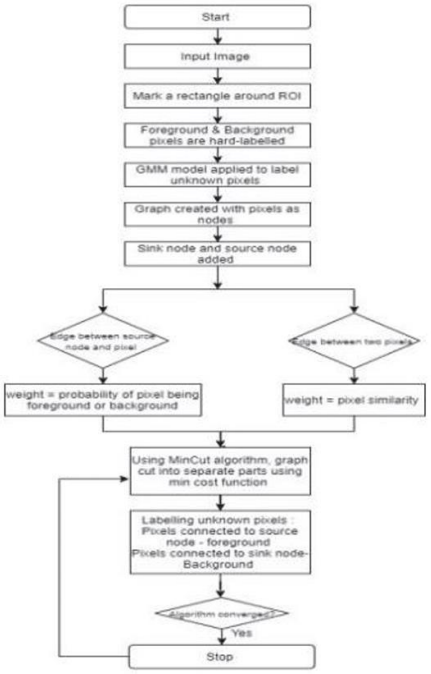
Software requirements:

- Operating System: Windows
- Coding Language: Python 3.7
- Script:
- Database :

Hardware Requirements:

- Processor - Pentium –III
- Speed – 2.4 GHz
- RAM - 512 MB (min)
- Hard Disk - 20 GB
- Floppy Drive - 1.44 MB
- Key Board - Standard Keyboard
- Monitor – 15 VGA Colour

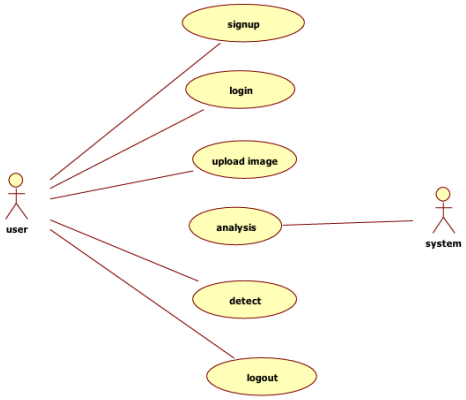
IV. SYSTEM ARCHITECTURE



V. SYSTEM DESIGN

UML Diagrams: The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

USECASE DIAGRAM: A use case diagram in the Unified Modeling Language (UML) is a type of behavioral 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.



VI. SOFTWARE ENVIRONMENT

What is Python?

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python :-

Let’s see how Python dominates over other languages.

Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

Extensible

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

Improved Productivity

The language’s simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

VII. SYSTEM IMPLEMENTATION

"text/plain": [

"<Figure size 432x288 with 1 Axes>"

]

},

"metadata": {

"needs_background": "light"

},

"output_type": "display_data"

}

],

"source": [

"sns.lineplot(data[\"N\"],data[\"label\"])"

]

},

{

"cell_type": "code",

"execution_count": 11,

"metadata": {},

"outputs": [

{"name": "stderr",

"output_type": "stream",

"text": [

"/home/user/test/lib/python3.6/site-packages/seaborn/_decorators.py:43:

FutureWarning: Pass the following variables as keyword args: x, y. From version

0.12, the only valid positional argument will be `data`, and passing other arguments

without an explicit keyword will result in an error or misinterpretation.\n",

" FutureWarning\n"

]

},

{

"data": {

"text/plain": [

"(array([-20., 0., 20., 40., 60., 80., 100., 120., 140., 160.]),\n",

" <a list of 10 Text xticklabel objects>)"

]

},

"execution_count": 11,

"metadata": {},

"output_type": "execute_result"

},

{

"data": {

},

{

"data": {

VIII SYSTEM TESTING

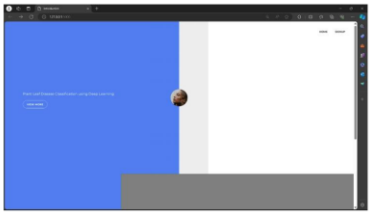
The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

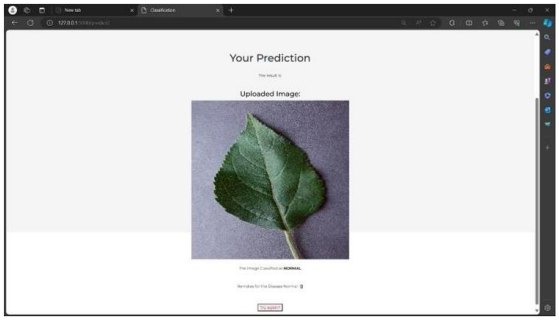
6.1 TYPES OF TESTS

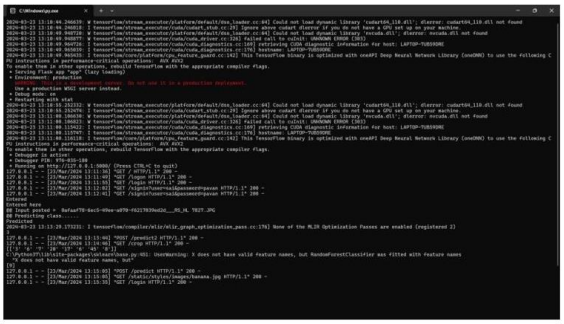
Unit testing Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

IX SCREENSHOTS







CONCLUSION

During our analysis, we have understood the need for efficient plant disease identification & classification algorithms and prevention methods. Due to a large number of crops and diseases available, it is crucial that the detection system should be able to adapt to the changing variables and trends. Hence, Machine learning and Deep learning approaches were employed for this project which ensures that the code trains itself against as many possible numbers of different crops and diseases as possible The paper consists of an android application covering plant disease detection and other functionalities such as language translation, weather forecasting and fertilizer calculator. With this application we aim to provide aid in the unprecedented agricultural activities and ensure a healthy plant. Through our thorough literature review and robust implementation, we have tried several approaches as discussed above and chosen the best model- CNN with accuracy of 97.94 using 20 epochs. We have also tested our application on cotton dataset and performed realtime analysis on a diseased tomato crop to ensure our model does not overfit and performs well in a live environment. In future, we aim to expand our dataset to include more varied types of crops and disease so that the algorithm can adapt better to real time conditions and provide wide coverage.

REFERENCES

[1] CropLife International (May 2015). India's farmers fighting pests. Retrieved from: https://croplife.org/news/keeping-indias-pests-in-line/

[2] Economic Times (Sept 2018). India sets record farm output target for 2018 19. Retrieved from:

[3] Sharada P. Mohanty David P. Hughes and Marcel Salathé."Using Deep Learning for Image-Based Plant Disease Detection."Front. Plant Sci., 22 September 2016

[4] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. 2004. "GrabCut": interactive foreground extraction using iterated graph cuts. In ACM SIGGRAPH 2004 Papers (SIGGRAPH '04). Association for Computing Machinery, New York, NY, USA, 309–314.

[5] R. Chapaneri, M. Desai, A. Goyal, S. Ghose and S. Das, "Plant Disease Detection: A Comprehensive Survey," 2020 3rd International Conference on Communication System, Computing and IT Applications (CSCITA),Mumbai, India, 2020, pp. 220-225, doi: 10.1109/CSCITA47329.2020.9137779.

[6] Raghavendra, B. K. (2019, March). Diseases Detection of Various Plant Leaf Using Image Processing Techniques: A Review. In 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), (pp. 313-316). IEEE.

[7] Malathi, M., Aruli, K., Nizar, S. M., & Selvaraj, A. S. (2015). A Survey on Plant Leaf Disease Detection Using Image Processing Techniques. International Research Journal of Engineering and Technology (IRJET), 2(09)

[8] Kaur, S., Pandey, S., & Goel, S. (2018). Semi-automatic leaf disease detection and classification system for soybean culture. IET Image Processing, 12(6), 1038-1048.