iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** V **Month of publication:** May 2023

DOI: https://doi.org/10.22214/ijraset.2023.51367

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue V May 2023- Available at www.ijraset.com

Plant Recognition and its Care using TfLite, Teachable Machine

Rashi Sahay[1], Ramanjot Kaur[2]
Apex Institute of Technology, Chandigarh University, Gharuan, Mohali (Punjab), India

Abstract: ***Plant identification is a significant undertaking however its care is considerably more significant. The identification of plants has a lot of benefits for wide range of people ranging from forestry services, pharmaceutical labs, government and common public. The research paper aims on developing a machine learning technology combined with mobile app development which is able to recognize plants and suggest good care for them like how much water, sunlight, fertilizer, etc is optimum for the plant. Plant identification with its care, based on leaf, is becoming an interesting task. Each plant leaf carries unique identity, which matches to no other plant, information that can be used in the identification and subsequently, determining measures for its care. A large number of features were taken from each leaf such as its length, area of hull, perimeter and color. All the work of research in this filed has further aggravated an interest in the development of automated systems for the recognition of different plant species. A full autonomous method of recognizing plants using Artificial Intelligence tools like Teachable Machine and TfLite. We gathered dataset from google images and several images with varying textures and lighting conditions clicked from my smartphone camera from the Kangra forest range. There are several applications that help in recognizing the plant species using features from their leaves but we are also adding an additional feature of its care.***

Index Terms: Plant Recognition, care, Tensorflow, Trainingthe model, faster_rcnn_inception model.

I. INTRODUCTION

Individuals see numerous sorts of plants in their daily life, some can distinguish them while others can't. For non-specialists, plant distinguishing proof is a troublesome assignment, further considering their highlights to realize its consideration is a lot of dreary. In addition to the fact that plants is a part and parcel of our planet Earth, however they likewise structure the very base of all nature and the food chain. To utilize and ensure plant species, it is vital to consider and group plants effectively. One route is to turn upward in the represented plant handbooks. The other path is to look on web or utilize versatile applications, for example, PlantNet, Plant-OMatic, and Wild Plant Database. So as to comprehend the coordinated outcomes, these administrations expect clients to give however much data as could reasonably be expected, for example, the plant's photograph, family name, geolocation, bloom hues, leaf shape, development structure, and so on. Discussing the name of plant species, even with the given data, we get a long competitor list to filter the information we need. Past these stages, a large number of the procedures engaged with grouping these plant species is 'human being abilities and accumulation of knowledge'. To execute the AI based plant distinguishing proof and care framework, building up a model dependent on a few basic highlights like leaf shape, veins, form, and so on is must. So, in this paper there will be utilization of these qualities as a contribution to the classifier that does this tedious task by utilizing AI calculations. The project is all about making a technology to combine the data science technology of machine learning with mobile application development. Machine learning is a very useful technology which is able to do recognitions just like human beings, when trained right. But the problem with machine learning is that it cannot be used by a user. To facilitate non complicated use of machine learning, it will be combined with mobile application so that the users can use it easily.

The dataset for developing a well-trained machine learning model will come from the local garden center, different plant species from Kangra forest range. The dataset would also be collected from various image results coming from Google search. The desired dataset for the project is going to be clean photos of plant leaves. The reason for choosing plant leaves as the basis for differentiation among the different plant species is that despite every plant's different shape and span, some can resemble up to 80% with other plant species which will be a problem for the training of the machine learning model. Since, all plant have leaves that differ by at least 70% from other plant species. Therefore, choosing plant leaves as the measure of classification is going to build a rigorous model. Next step would be clustering of the data to divide it into train and test datasets followed by the rigorous training to make a well performing machine learning model.

TensorFlow library, OpenCV are going to be used in training. To convert the machine learning model into phone usable form, Google's tool, Teachable Machine, is useful. Teachable machine is a suite of tools that is used to train machine learning models by providing the dataset to it. It can help train gesture detection, audio recognition and TensorFlow lite models. The choice of using TensorFlow models comes from its special feature where it can be used in a smartphone application. TensorFlow is a Google provided library for training computer vision neural network models with numerous layers in it. TensorFlow was first developed by Google Brains team for their internal usage which was later rolled to production by the team. This TensorFlow model would then provide us with a trained model file with .tf extension to be used in the mobile application. TensorFlow Lite is a suite of tools that facilitates programmers to combine the good of machine learning with the versatility of mobile applications.[1] It is useful in many ways like it is optimized for machine learning getting optimistically utilized by the machines, by addressing 5 key constraints: privacy, latency, another big advantage is that it requires no internet and time lag for performing. It is a multiple platform support, covering Android and iOS devices, embedded Linux, and microcontrollers. Diverse languages like Java, Kotlin, Flutter, Swift can be used with it. Various examples of attaching the machine learning based models with mobile applications have already been developed using TfLite.[1] This TfLite model file would then be used in the backend of the mobile application. Mobile application would take real-time input feed from the camera do the predictions of the plant name[13]. The complete data related to the requirements of the identified plant would be saved in the backend. The data would be retrieved making HTTP request to the backed server to display the information in real-time to the user.

II. LITERATURE REVIEW

Several examinations have been done to develop tools for the identification of plants in the last 10-20 years. In 2007, Shu Shan proposed the machine learning system which increased the properties for certain features. The features inherited in this research work were shape and veins. He used the Probabilistic Neural Network for the training ofhis model which got an accuracy level of 90.3%.[2,14]

In 2012, Abdul Kadir built up a plant identification methodology for 55 examples of leaves. It was created to perceive any plants having similar shapes yet a big difference in the leaf color features. Zernike moments were utilized consolidating with different highlights, instance shading, geometric, and Gray-Level Co-occurrence Matrix (GLCM) .[3]

In 2013, Amin and Khan carried out a research on the leaf curvature to be able to add to the formerly proposed plant leaf feature extraction technique. They used the k-nearest neighbour Algorithmand ended up at an accuracy of 71.5%.[4,15]

In 2014, Hernandez-Serna and Jimenez-Segura were able to end at an accuracy level of 92.9% utilizing the Flavia dataset. 16 (6 geometrical, 8 surface and 2 morphological highlights) were taken care of to an Artificial Neural Network (ANN) 60 nodes in the hidden layer and a learning rate of 0.1 over 50000 generations. Utilizing the equivalent dataset, Chaki et al. (2015) got a great accuracy too. Utilizing shape includes just on the Flaviadataset and Pattern Net (a kind of neural system), Siravenha and Carvalho arrived at a comparative exactness as Chaki et al. Their neural network had 26 layers of neurons and more than 100 epochs.[5]

Utilizing a k-nearest neighbour (kNN) classifier, Munisami etal. (2015) got an astonishing accuracy of more than 86 % by taking samples of 640 leaves from 32 species of different plants. They utilized shape and shading data as it were. The pictures were procured utilizing a cell phone camera with a goals of 1980*1024.[6,16]

A fascinating work was finished by Carranza-Rojas and Mata-Montero (2016) in which they developed a clean and a noisy dataset. They executed the Histogram of Curvature over Scale (HCoS) algorithm to extract the information about the contour andthe local binary pattern variance (LBPV) to extract texture information. In the best case, the dataset that was clean beat the noisydataset by 7.3%.[7,17] This propose pictures taken from a smartphone camera can deliver agreeable degrees of accuracy compared with pictures which are physically prepared in a lab and afterward characterized. Earlier, Amin and Khan (2013) have used an appropriated progressive diagram neuron (DHGN) to catch shape data utilizing 64 element vectors and the k-closest neighbor classifier with Canberra distance to acquire a precision of 71.5%.[8,18]

Table 1.1 Training Dataset

S. No.	PLANT NAME	INSTANCES
1.	Rose	59
2.	Mint	61
3.	Bamboo	03

III. METHODOLOGY

For this research two datasets were used. One is collected from the publicly available images of the required plants using Google image search. Another is dataset made by collecting images from our locality using the smartphone camera. Pre-processing is performed by converting BGR images to RGB system to generate a good numpy array for it to improve the accuracy. Then, it generates the tensorflow records for the images as per labeling of each plant leaf in the image. By applying the technique of feature extraction to extract vein feature, texture of leaf and morphological features of the leaf. Then by saving these features as feature matrix which are then to be fed to the classifier.

For DAVIS and Pascal VOC datasets, containing more than a thousand pictures with different scenes, which are more appropriate for viable undertakings. The strategy accomplishes the best outcomes contrasted and different techniques. Investigating the outcome,

It is considered that early-late combination procedure can forestall highlights of intelligent snaps weakening over the organization.[9] Subsequently, the technique answers the snaps effectively and is hearty in different scenes mIoU per click. There are three perceptions: 1) Our technique has the steadiest presentation contrasted and different strategies, where mIoU improves bit by bit as the snaps increments. In any case, different strategies have the debasement issue. For instance, mIoU of RITM diminishes drastically when the snaps increment from 3 to 5, and 9 to 11 on GrabCut dataset. As portrayed previously, the proposed edge assessment can work on the division security. 2) This technique requires less snaps to accomplish higher mIoU contrasted and different strategies. In the Berkeley dataset, NoC@95 of the used strategy is about 4, while others are more than 10. 3) Using each of the 20 ticks, this strategy still accomplishes the most elevated mIoU on all datasets. The examination exhibits the prevalent division capacity of our technique.

IV. DATASET ACCESSION

Dataset refers collection of standard set of images used for a particular project development. In this case of work, there are two different kinds of datasets (1) collected from google images (2) Local dataset. A. Dataset collected from google images: This dataset consists of a collection of images of the plants namely Mint, Rose, Bamboo, Curry leaf. B. Dataset collected by clicking local images from the smartphone camera. All the images in this dataset are of JPG format as the train.py uses only JPG images as they are good for training purpose. Here is the table to show the composition of this dataset:

Table 1.2 Test Dataset

S. No.	PLANT NAME	INSTANCES
1.	Curry Leaf	71
2.	Rose	56
3.	Mint	59
4.	Bamboo	75

Fig. 1 Test dataset for Machine Learning

V. SETTING UP ENVIRONMENT

An environment needs to be setup for this project as per the hardware and software requirements of the deep learning algorithms and smooth functioning of the libraries we use. The steps for setting up the environment in this paper are as:

Installation and Downloading: Here it is needed to install the Anaconda and then various libraries that we are to use for our project including the repositories of some open-source projects from www.github.com and the model file from the official model zoo of tensorflow. Here is a table for various github commits

Certain changes: There have been made a number of changes in the downloaded repositories as per the requirement because the repositories that are downloaded are clone of some other open-source projects that contain various things that are of need to start from anew.

Setting up virtual-environment: In this, there is a set-up of a separate virtual environment for our paper that contain the suitable python version with the suitable version of other libraries that we need.

In this work, we foster an intuitive division apparatus

utilizing the proposed model. Our instrument intends to assist clients with clarifying division datasets effectively and precisely. The comment pipeline incorporates three stages: information readiness, intuitive explanation and polygon outline altering.

During information planning, our apparatus upholds information designs in various areas including regular pictures, clinical imaging, remote detecting pictures, etc. After the picture is stacked into the product, it tends to be zoomed, moved and pre-handled by changing brilliance and differentiation. The explanation cycle is exceptionally adaptable and configurable. Most tasks in the application support console alternate routes which can be changed by the client.

During intelligent comment, clients add positive and

negative focuses with left and right mouse clicks, separately. The application runs model deduction and shows the client forecast outcome. The client can change the objective boundary by changing the limit to recognize forefront and foundation pixels to get more exact division results. The device additionally upholds separating the biggest associated area.

This element is helpful when there are different focuses of the same sort in a picture. Stifling little sure locales can liberate clients from clicking negative focuses in each of these areas.

Fig. 2 Train

Subsequent to completing intelligent division, the instrument creates a polygon outline around the objective line. Clients can change the polygon vertexes to additionally further develop division exactness. It is adaptable for reasonable division undertakings.

At times, changing the polygon edge could be quicker than adding many snaps during intelligent division, with the goal that it further develops the general comment effectiveness. At last, the division results can be saved in numerous organizations including division cover, PASCAL VOC, COCO, pseudo tone, etc.

Countless named picture information are typically fundamental for division models. Because of the significant expense of the pixel-level explanations, intuitive division turns into a proficient method for separating the object of interest.

In this work, we propose a clever intelligent engineering named EdgeFlow that completely uses the client communication data with no post-handling or iterative advancement plot. With the coarse-to-fine organization plan, our proposed technique accomplishes cutting edge execution on normal benchmarks. Moreover, we foster an effective intuitive division device that helps the client to

further develop the division result continuously with adaptable choices. In future work, we will deal with lightweight models that can be sent into different stages. Another promising point we are dealing with is multi-methodology, which uses more info types, similar to sound and text. The unique sorts of data sources can complete one another, with the goal that it is fundamental to further develop the quality further.

Verification: We need to verify if the dependencies have been installed correctly or not. So we compile the protobuf and hence, check if the object detection API provided by the Tensorflow compiles successfully or not.

VI. GATHERING AND LABELLING DATA

Gathering: Gathering of data consist of the process to collect datasets from different sources like www.kaggle.com , www.github.com, etc. that consist ofimages of the plants that we work on for care and recognition.
These Convolutional Neural Networks work as a system of layers of the neural network present in the
nervous system of the animals. Each neuron carries certain small amount of information which is further linked to several other neurons and facilitate the model to learn and train.
Labelling: Labelling for different parts of the images which have different evidences of containing different plants in it. We use LabelImg annotation tool to draw rectangular boxes around the plant parts and then label them according to their common names.[10]

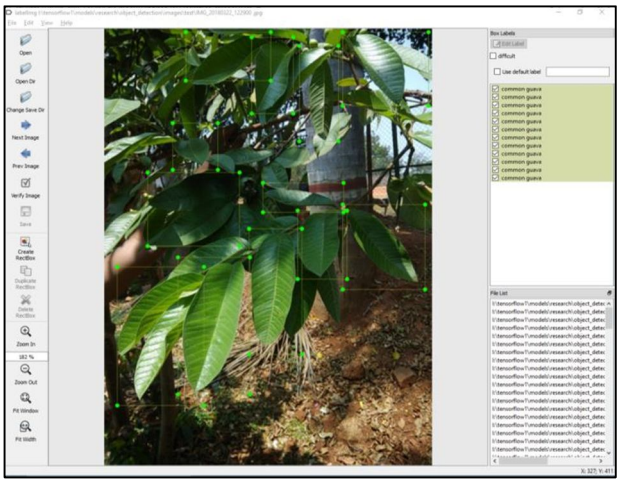


Fig. 3 Segmenting the dataset

VII. TRAINING

Labelmap File: For re-training the pre-trained faster_rcnn_inception model according to our dataset, we need to generate a label map file that contains the names of the plant species that we are recognizing in the form of a particular syntax, with each plant species getting a particular index value.[11] The syntax to follow while writing the label map file is:

```
item {
id: 1
name: 'rose'
}
item {
id: 1
name: 'mint'
}
item {
id: 1
name: 'bamboo'
}
```

Fig. 4 Labelmap file for training

Training: Running the trainer for our dataset consist of changing the input paths in the original train script and other supportive scripts according to the requirements ofour system and then providing the model we use for training as argument.
The training script for the plant detector is designed as a Convolutional Neural Network.
These Convolutional Neural Networks work as a system of layers of the neural network present in the
nervous system of the animals. Each neuron carries certain small amount of information which is further linked to several other neurons and facilitate the model to learn and train.

VIII. RUNNING THE CLASSIFIER

The final part of the project will contain a script that runsthe Plant Detector Classifier which will work for (i) feed from an image (ii) feed from a video (iii) feed from a live_webcam.[12] The detector recognizes the plant in the input by using the inference_graph obtained in the training part. We get a rectangular box drawn around the plant that gives the score of confidence with the name of the species. Our classifier contains a special feature of also giving a tip for the care of theidentified plant.

IX. RESULTS AND DISCUSSION

Here we discuss about the experimental results of our work. The total loss during the training starts at a high value and then keeps on dropping down till it becomes constant at a considerably lower value. Using the faster_rcnn_inception model the training loss starts at approx. 2.0 and then quickly drops down to 0.8. After this value, the loss value experiences many ups and downs as its value drops considerably slow than before.

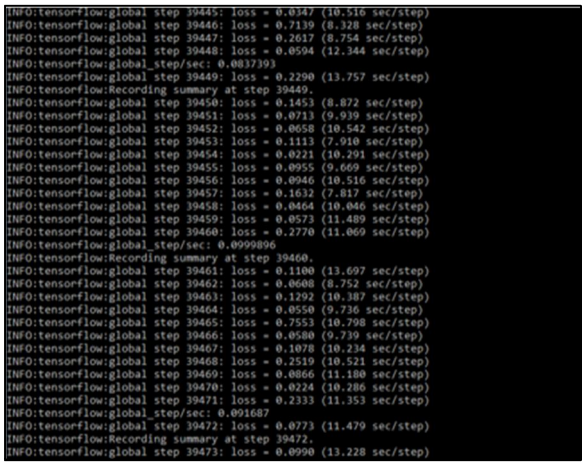


Fig. 5 Training the dataset

Fig. 6 Result after training

We wait until the loss becomes steady at a value of 0.05 which takes about 45,000 steps. The time duration took for training the whole model was about eight days, depending upon how powerful your CPU/GPU is.

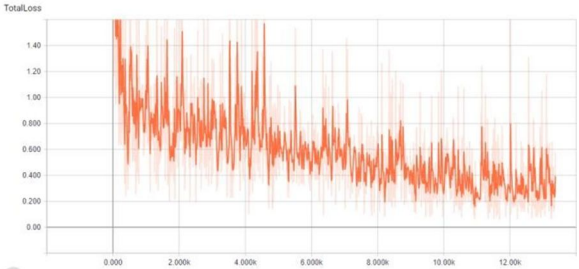


Fig. 7 Total loss Graph

Re-training the same model with ssd_mobilenet model pre-trained model, instead of faster_rcnn_inception model takes a considerably larger time and does not match the level of accuracy and robustness of faster_rcnn_inception model. Ssd_mobilenet model is useful for low power machines like smartphone, webcam, Rasberry pi, etc while the faster_rcnn_inception model works good for high power machines.

We got the feature matrix and the confusion matrices for our work as follows:

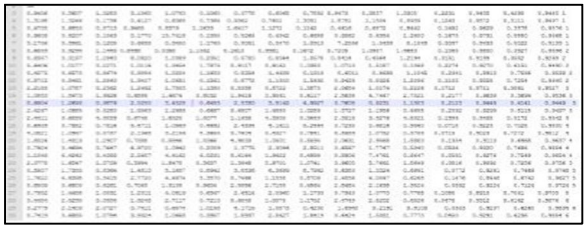


Fig. 8 Feature Matrix

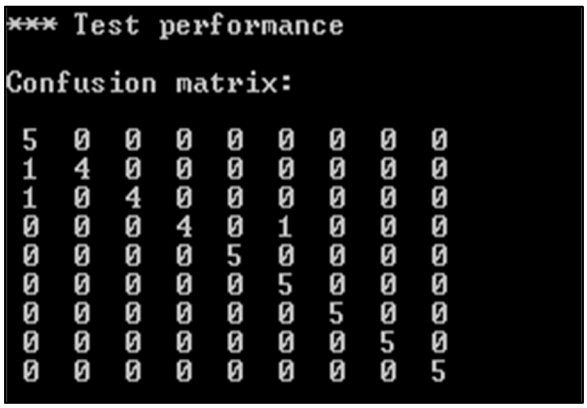
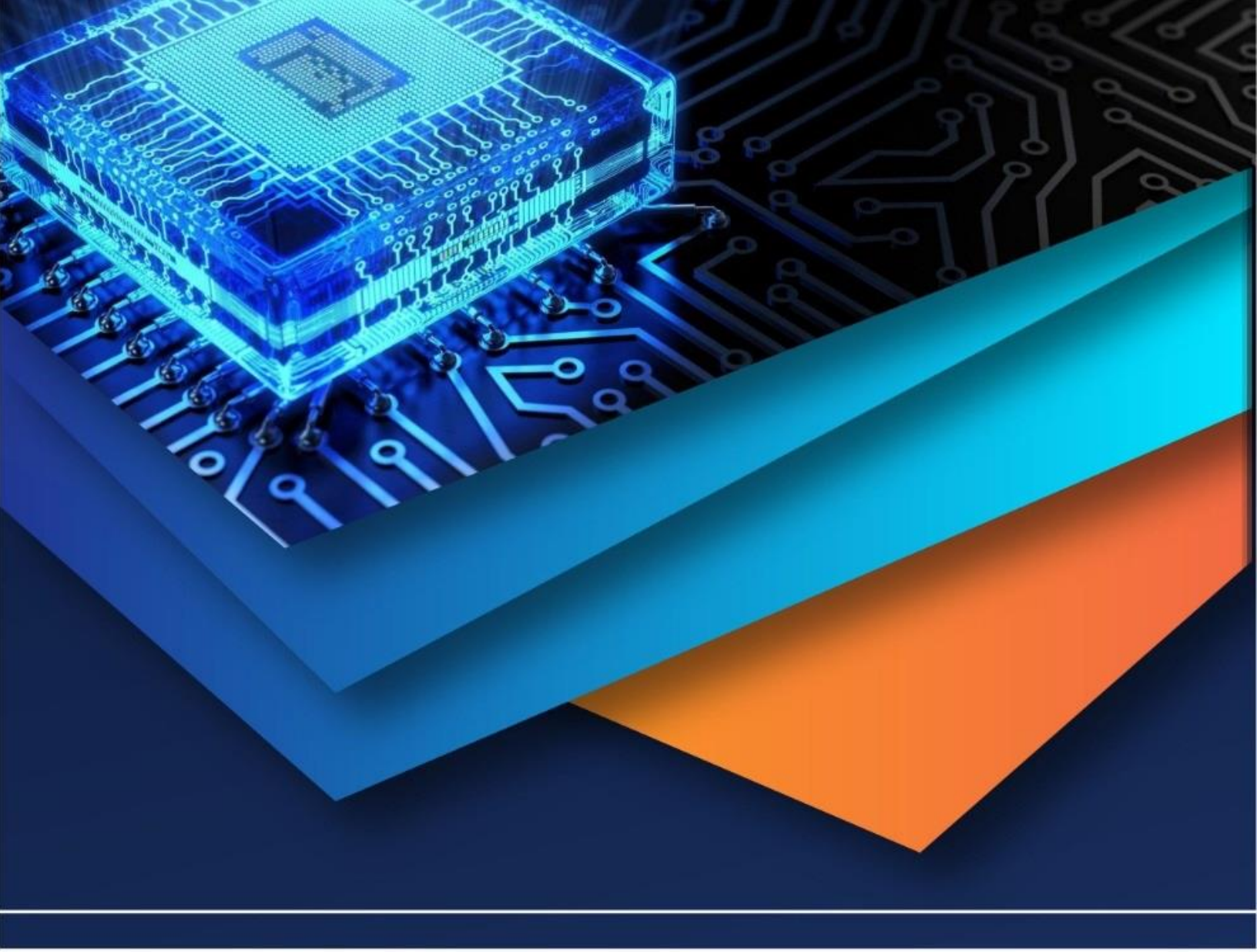


Fig. 9 Confusion Matrix

REFERENCES

[1] Shafique, M., Theocharides, T., Reddy, V. J., & Murmann, B. (2021, December). TinyML: current progress, research challenges, and future roadmap. In 2021 58th ACM/IEEE Design Automation Conference (DAC) (pp. 1303-1306). IEEE.
[2] Detection of Plant Leaf Diseases using Image Segmentation and Soft Computing by Vijai Singh A. K. Mishra
[3] Guide to Plant Collection and Identification by Jane M. Bowles PhD
[4] A Comprehensive Guide to Convolutional Neural Networks – the ELI5 way by Sumit Saha
[5] Neural Networks by Taro Toyoizumi, DeLiang Wang
[6] Deep Learning in Neural Networks: The science behind an Artificial Brain by Sarat Kumar Sarvepalli
[7] An Introductory Review of Deep Learning for Prediction Models with Big Data by Frank Emmert, Zhen Yang, Han Feng
[8] Machine Learning for Mobile Developers: TensorFlow Lite Framework by Avid Far Hoodfar
[9] TensorFlow Lite: On-Device Machine Learning Framework by Li Shuangfeng
[10] Automatic Recognition of Medicinal Plants using Machine Learning Techniques by Adams Begue, Venitha Kowlessur.
[11] Zhihan Lv, Liang Qiao et al. 2021. AI-enabled IoT-Edge Data Analytics for Connected Living. ACM Trans. Internet Technol. 21, 4, Article 104 (November 2021), 20 pages.
[12] Dhaka VS, Meena SV, Rani G et al. "A Survey of Deep Convolutional Neural Networks Applied for Prediction of Plant Leaf Diseases", Sensors. 2021; 21(14):4749.
[13] Loveleen Gaur, Gurmeet Singh, Arun Solanki et al., "Disposition of Youth in Predicting Sustainable Development Goals Using the Neuro-fuzzy and Random Forest Algorithms", in HCIS, Springer, (2021)
[14] Arora, M.et al. (2020). A Systematic Literature Review of Machine Learning Estimation Approaches in Scrum Projects. In: Mallick, P., Balas, V., Bhoi, A., Chae, GS. (eds) Cognitive Informatics and Soft Computing. Advances in Intelligent Systems and Computing, vol 1040. Springer, Singapore.
[15] Monica Sood et al. "Optimal Path Planning using Swarm Intelligence based Hybrid Techniques" in JCTN. Vol. 16 No. 9, 2019, pp. 3717–3727
[16] M. Kumar et al. "Improved Deep Convolutional Neural Network based Malicious Node Detection and Energy-Efficient Data Transmission in Wireless Sensor Networks," in IEEE Transactions on Network Science and Engineering



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY