iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** X **Month of publication:** October 2026

DOI: https://doi.org/10.22214/ijraset.2026.85468

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 14 Issue X Oct 2026- Available at www.ijraset.com

PhytoSense AR: Augmented-Reality Plant Disease Diagnosis Using a Field-Adapted Vision Transformer and Colour-Based Lesion Segmentation

Sonu Airen, Puja Gupta, Yameesh Nayak, Suyash Gupta, Swapnil Pal, Neha Agrawal
Department of Information Technology Shri G.S. Institute of Technology and Science, Indore Indore, India

Abstract: ***Plant diseases cause substantial crop losses, and smallholder farmers often lack timely access to expert diagnosis. Deep-learning classifiers trained on the PlantVillage dataset report near-perfect accuracy, yet they degrade sharply on photographs captured in the field. This paper presents PhytoSense AR, a free, browser-based diagnosis system that overlays disease information directly on the live camera view. The systemcombinesa colour-spaceleafandlesionsegmentationstage,basedonHSV thresholdsandtheExcess-Green index, witha VisionTransformer(ViT-B/16) classifier covering 38 classes across 14 crops. To narrow the gap between laboratory and field imagery, the classifier was fine-tuned on a mixture of PlantVillage and the PlantDoc field dataset with augmentation that simulates changes in zoom, orientation, illumination and focus, and was then exported to an 8-bit quantised ONNX model of 87 MB that runs on a CPU without PyTorch. On the PlantDoc test set of 236 unseen field images, top-1 accuracy rose from 37.7 % for the originalPlantVillage-onlymodelto76.7%forthedeployedquantisedmodel,whileaccuracyon760held-out PlantVillage images remained at 99.9 %. End-to-end processing, including segmentation and classification, takes about 200 ms per image on a laptop CPU. The system also estimates an infected-area severity and a health score, provides treatment guidance from a curated knowledge base, and offers a multilingual assistant. The results show that domain-adaptive fine-tuning on a modest field dataset, combined with lightweight deployment, makes AR-assisted plant diagnosis practical on commodity hardware.***

Index Terms: ***plant disease detection, vision transformer, augmented reality, domain adaptation, model quantisation, precision agriculture***

I. INTRODUCTION

A. Background

Agriculture remains the principal source of livelihood for a large share of India's population, and plant diseases are one of the most persistent threats to its productivity. A global expert survey estimated that pathogens and pests reduce the yields of wheat, rice, maize, potato and soybean by roughly 17–30 % [1], with the heaviest losses in food-deficit regions. Early and correct identification of a disease decides whether a farmer applies the right treatment in time, applies an unnecessary chemical, or loses the crop. Expert diagnosis through agricultural extension officers is accurate but slow and unevenly available, especially for smallholders in rural areas.

The wide adoption of smartphones has made image-based diagnosis a realistic alternative. The release of the open PlantVillage image collection [2] enabled a series of deep-learning studies that report accuracies above 99 % for leaf-disease classification [3], [4]. These results created strong interest in mobile diagnostic tools that a farmer can use by simply photographing a leaf.

B. Problem Statement

Mostpublishedclassifiersaretrained andtested onPlantVillage, whose imagesshow asingle detached leafona uniform background under controlled lighting.

Several studies have shown that models trained on such images lose muchoftheiraccuracy onphotographstakeninrealfields, wherebackgroundsarecluttered,severalleavesoverlap, and lighting and focusvary [3]–[6].

©IJRASET: All Rights are Reserved

The present project encountered exactly this behaviour: an early prototype that used a publicly available PlantVillage-trained Vision Transformer achieved only 37.7 % top-1 accuracy on the PlantDoc field test set, although it classified the laboratory samplesalmost perfectly. In addition, typical classifiers return only a label; they do not show the user *where* the symptoms are or *how much* of the leaf is affected, andmany prototypes depend on GPUs or paid cloud services that are impractical for farmers and for student projects.

C. Motivation

A useful field tool must therefore satisfy four conditions at once: it must work on real photographs, explain its result visually, give actionable advice, and run at no cost on ordinary hardware. Augmented reality (AR) offers a natural way to explain a result, because the diagnosis can be drawn directly over the live camera image of the leaf. Recent advances in Vision Transformers [7] and post-training quantisation [8] make it possible to run an accurate model on a CPU, which removes the need for dedicated hardware.

D. Objectives

The objectivesoftheprojectare: (i) to identify38diseaseandhealthy classesacross14 crops from aleafimage;
(ii) to localise the leaf and its lesions and present them as a live AR overlay; (iii) to estimate the infected-area severity and an overall health score; (iv) to provide treatment and prevention guidance and an interactive multilingualassistant;and(v)toachieveallofthiswithfree,open-sourcecomponentsthatrunonalaptopCPUora free cloud instance.

E. Contributions

Themain contributionsofthisworkareasfollows:

- An end-to-end, zero-cost diagnosis system, PhytoSense AR, delivered as an installable Progressive Web App (PWA) that overlays the leaf outline, lesion regions and diagnosis on the live camera view.
- A training-free colour-space segmentation stage that combines HSV thresholds, the Excess-Green index and a border-based background model to localise the leaf and its lesions, from which an infected-area severity and a health score are derived.
- Field-adaptive fine-tuning of a ViT-B/16 classifier on PlantVillage and PlantDoc [6], with an explicit mapping of PlantDoc classes onto the 38-class taxonomy, which raised field accuracy from 37.7 % to 76.7 % without loss of laboratory accuracy.
- A CPU-only deployment based on an 8-bit quantised ONNX model (87 MB, about 200 ms per image), small enough for a 512 MB free cloud instance.
- Atransparentevaluationthatreportsper-classfieldaccuracy andstatesthelimitationsoftheapproach.

F. Paper Organisation

Section IIreviewsrelatedworkandidentifiestheresearchgap. SectionIIIdescribesthe proposed systemand its algorithms. Section IV covers the implementation. Section V presents and discusses the results, and Section VI concludes the paper and outlines future work.

II. LITERATURE REVIEW

A. Classifiers Trainedon Laboratory Images

Mohanty*etal.*[3]trainedAlexNetandGoogLeNeton54,306PlantVillageimagesof38classesandreached
99.35 % accuracy on a held-out split. The same work showed that accuracy dropped to about 31 % on images collected from other sources, an early warning about generalisation. Ferentinos [4] evaluated several CNN architectures on87,848imagescovering58plantdiseasecombinations,includingsomefieldimages,andobtained99.53%withVGG;thestudyalsoreportedasharpfallinaccuracywhenmodelstrainedon laboratoryimageswere tested on field images. Too *et al.* [9] compared the fine-tuning of VGG-16, Inception-v4, ResNet and DenseNet on PlantVillage andfoundthatDenseNet-121offeredthebestaccuracywithfewerparameters.Atila*etal.*[10]applied EfficientNet variants to Plant Village and reported accuracies above 99 % on both the original and an augmented version. These works established strong baselines, but all of them were evaluated mainly on images from the same laboratory distribution.

B. Generalisation and Field Data

Barbedo [5] analysed the factors that limit deep learning for plant disease recognition and identified the lack of diverse, field-representative training data, background clutter, and symptom variability as the main obstacles. In a follow-up study, Barbedo [11] classified individual lesions and spots instead of whole leaves, which increased the effective size of the dataset and improved accuracy, but required lesion-level annotation. Singh *et al.* [6] released PlantDoc, a datasetof about2,600 images of 13 plant speciesand 27 classes gathered from real-world sources, and demonstrated that models trained only on PlantVillage perform poorly on field images while fine-tuning on PlantDocimprovesthemconsiderably.Li*etal.*[12]revieweddeep-learningmethodsforplantdiseasedetectionand classification and highlighted small field datasets, early-stage symptom detection and robustness to imaging conditions as open problems. Ahmad *et al.* [13] surveyed more than one hundred studies and recommended that future tools be validated on field images and designed for the needs of end users.

C. MobileandTransformer-BasedApproaches

Ramcharan*et al.* [14] deployed a MobileNet-based object detector for cassava diseases on a smartphone and observed that performance on real-world video was lower than on curated test images, which underlines the difficulty of in-field use. Borhani*et al.* [15] compared Vision Transformer, CNN and hybrid models on several plantdiseasedatasetsandreported thatattention-basedmodelscanmatchorexceedCNNaccuracy,withatrade-off ininferencespeed. Thakur*etal.* [16]proposed PlantXViT, alightweightcombinationofconvolutionalblocksanda Vision Transformer with explainability, aimed at deployment on resource-limited devices. Table I summarises the reviewed works.

TABLEI
SUMMARYOFREVIEWEDWORKS

Ref.	Approach	Data	Mainlimitation
[3]2016	AlexNet,GoogLeNet	PlantVillage	llsto~31%onother sources
[4]2018	VGGandother CNNs	87,848lab+fieldimages	lmodelsweakonfield images
[5]2018	Analysisoflimitingfactors	Multiple	Nodeployablesystem
[9]2019	Fine-tunedDenseNet,ResNet	PlantVillage	Laboratoryimagesonly
[11] 2019	Lesion-levelCNN	Ownfieldlesions	Needslesionannotation
[14] 2019	MobileNetdetectoronphone	Cassavafieldimages	;loweraccuracyon video
[6]2020	dataset,CNNfine-tuning	~2,600fieldimages	Smalldataset;noapplication
[10] 2021	EfficientNetB0–B7	PlantVillage	Laboratoryimagesonly
[12] 2021	Review	—	Identifiesopenproblemsonly
[15] 2022	ViT,CNNandhybrids	Severalpublicdatasets	:uracytrade-off;noAR or advice
[16] 2022	PlantXViT(CNN+ViT)	Severalpublicdatasets	Nolocalisationorseverity
[13] 2023	Survey	—	Callsforfield-validatedtools

D. Research Gap

The reviewed literature reveals three gaps. First, most high-accuracy models are validated only on laboratory images, and the few field-oriented studies seldom deliver a working application. Second, classification-onlysystems do not localise symptoms or quantify severity, so the user cannot verify the result visually. Third, deployable tools frequently rely on GPUs, paid APIs or a single crop. PhytoSense AR addresses these gaps by combining field-adaptive fine-tuning, training-free lesion localisation with severity estimation, AR presentation, a treatment knowledge base and CPU-only, zero-cost deployment in one system that is evaluated on an independent field test set.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue X Oct 2026- Available at www.ijraset.com

III. PROPOSED SYSTEM AND METHODOLOGY

A. System Architecture

PhytoSense AR follows a three-tier client–server architecture (Fig. 1). The client tier is a PWA written in plain JavaScript that captures camera frames, draws the AR overlay on an HTML canvas and shows results, history, the disease library and the assistant. The application tier is a Python FastAPI service that performs segmentation, classification, scoring and knowledge lookup and exposes a REST API. The data tier holds the ONNX model, a JSON knowledge base and an SQLite database of past scans; external services (Wikipedia and, optionally, Google Gemini [17]) are used only for supplementary information.

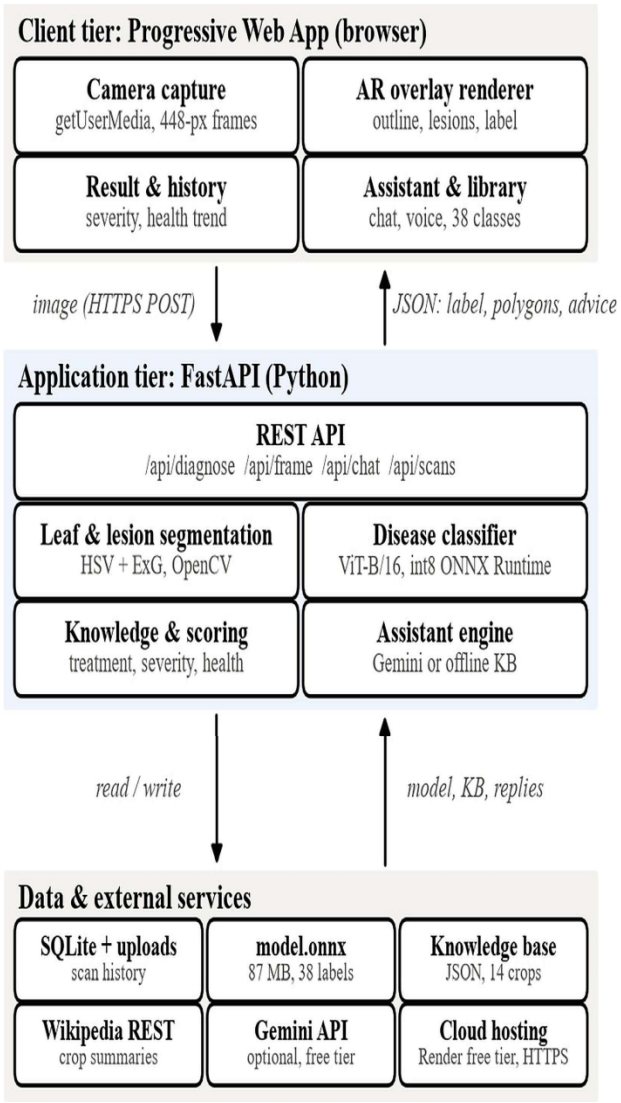


Fig.1.Three-tierarchitectureofPhytoSenseAR.ThebrowsersendsimagesoverHTTPS;theFastAPIservicereturnsthe diagnosis together with leaf and lesion polygons for the AR overlay.

B. Workflow

The system operates in two modes. In *live AR mode*, the client downscales the camera frame so that its longer side is 448 pixels, encodes it as JPEG and posts it to /api/frame; the next frame is sent as soon as the previous responsearrives,whichyieldsabouttwotothreeoverlayupdatespersecond ona laptop CPU. These framesarenot stored. In *capture mode*, the shutter sends the full-resolution photo to /api/diagnose; the full result, including the photo, is saved to the history. Both modes run the same pipeline, shown in Fig. 2.

©IJRASET: All Rights are Reserved

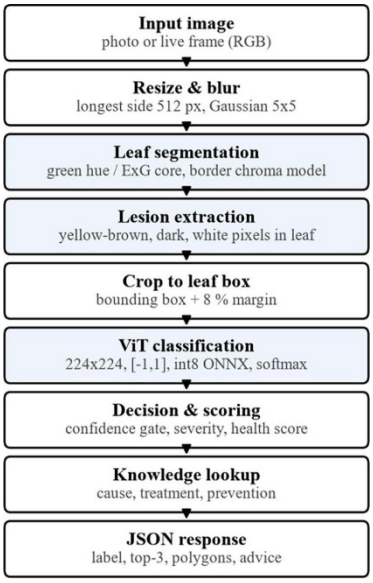


Fig.2.Per-imagediagnosispipeline.Shadedstagesperformthecomputer-visionanddeep-learningsteps;theremainingstages handle pre-processing, decision logic and response construction.

C. LeafandLesionSegmentation

The segmentation stage localises the leaf and its lesions without any trained model. The image is resized so that its longer side is at most 512 pixels and smoothed with a 5 × 5 Gaussian filter. Healthy tissue is detected in HSV space (OpenCV hue range 0–179) as pixels with a green hue of 33–95 that are either sufficiently saturated and bright or have a clearly positive Excess-Green index [18]:

$$ExG = 2G - R - B, \text{green if } ExG > 25. \quad (1)$$

The largestconnectedgreen region, after morphologicalopening, formstheleaf core;images whosecore covers less than 2 % of the frame are reported as *no leaf*. To include blighted margins without absorbing a brown table or soil, a background model is estimated from a six-pixel image border: the median *a\ and* b\ values in CIELAB are computed, and any pixel whose chroma lies within a distance of 9 from that median is treated as background-like. Symptom-coloured pixels (yellow, tan, brown, rust and purple-red hues) that touch the core and are notbackground-like are added to the leaf. If this growth exceeds three times the core area or reaches the image border on several sides, the hole-filled core alone is used. Inside a slightly eroded leaf mask, lesion pixels are those with symptomtones,darknecroticvalues(V<60)orwhitepowderycoating(lowsaturation, V>175)thatarenotgreen. After morphological opening and closing, contours smaller than 0.05 % of the leaf area are discarded and up to 40 lesion contours are simplified to polygons. The infected-area severity is

$$S = \min(100, 100 \cdot A_{\text{lesion}} / A_{\text{leaf}}), \quad (2)$$

where$A_{\text{lesion}}$and $A_{\text{leaf}}$are the pixel areas of the lesion contours and of the leaf. The procedure is summarised in Algorithm 1.

Algorithm1:Leaf andlesion segmentation

Input:RGBimage$I$ **Output:**leafpolygon,lesionpolygons,bbox,S

1:I←resize(I,maxside512);I←GaussianBlur(I,5×5) 2:G ←
green hue ∧(saturated∧bright ∨ExG> 25)
3:*core*←largestcomponent(open(G))
4:**if**area(*core*)<2%ofimage**thenreturn**noleaf 5:*bg*←
|(a, b) − median of border (a, b)| < 9
6:*grown*←componentsof close(G ∨(tone ∧¬*bg*))touching*core*
7:**if***grown*>3·*core***or**touches≥2borders**then***leaf*←fill(*core*) 8:**else**
leaf ← fill(*grown*)
9:*les*← (tone∨dark ∨white)∧¬G∧erode(*leaf*)

D. Disease Classification

Theoriginalimageiscroppedtotheleafboundingboxwithan8%margin,sothattheclassifierseesmainlyleaf tissue, and is resized to 224 × 224 pixels with bilinear interpolation. Each channel value $x \in [0, 255]$ is mapped to $[-1, 1]$:

$$x' = x/127.5 - 1. \quad (3)$$

TheclassifierisaViT-B/16VisionTransformer[7]with12encoderlayers,ahiddensizeof768and12attention heads. The image is split into 196 patches of 16 × 16 pixels, which are linearly embedded, combined with a learnableclasstokenandpositionembeddings,andprocessedbyself-attention.Theclass-tokenoutputismappedto logitszkfor the K = 38 classes and converted to probabilities by the softmax function:

$$p_k = \exp(z_k)/\Sigma_j \exp(z_j), k = 1, \ldots, K. \quad (4)$$

The three most probable classes are returned. The starting checkpoint is a publicly available ViT-B/16 that had been fine-tuned on PlantVillage; Section III-E describes how it was adapted to field images.

E. Field-AdaptiveFine-Tuning

*Data.*Twodatasetswere combined(TableII).FromPlantVillage[2],up to250imagesper classwere drawn for training, and a disjoint set of 20 images per class was held out to test whether laboratory accuracy is retained. PlantDoc [6] supplied real-field photographs; 10 % of its training split was reserved for validation, and its official test split was used only for the final evaluation. Each of the 28 PlantDoc training classes was mapped to the equivalent label of the 38-class taxonomy (e.g. *Bell_pepper leaf spot → Pepper, bell — Bacterial spot*, *Corn leaf blight → Corn — Northern Leaf Blight*), so the application's knowledge base and interface remained unchanged. BecausePlantDoc is small,each field imagewaspresented fourtimesper epoch, eachtime with a differentrandom augmentation. All images were downscaled once so that the longer side was 384 pixels, which reduced decoding time during training.

TABLE II
DATAUSEDFORFINE-TUNINGANDEVALUATION

Subset	Images	Use
PlantVillagetraining	9,382	Training(≤250perclass)
PlantVillageheld-out	760	Test,laboratory(20perclass)
PlantDoctraining	2,108	Training(×4perepoch)
PlantDocvalidation	234	Modelselection
PlantDoctest(official)	236	Test,fieldimages
Samplesperepoch	17,814	9,382+4×2,108

*Augmentation.*Training images were transformed by random resized cropping (scale 0.25–1.0, aspect ratio 0.7–1.4)to simulate partial leavesand varying distance,horizontalandverticalflips,rotation of upto±25°, colour jitter (brightness and contrast 0.4, saturation 0.3, hue 0.03) to simulate sunlight, shade and different phone cameras, and Gaussian blur (σ = 0.1–2.0, probability 0.3) to simulate out-of-focus shots. Validation and test images were only resized, exactly as in the deployed application.

*Optimisation.*The network was trained end-to-end with the cross-entropy loss with labelsmoothing [19], which discourages over-confident predictions on the noisier field labels:

$$\mathcal{L} = -\Sigma_k q_k \log p_k, q_k = (1-\varepsilon) y_k + \varepsilon/K, \quad (5)$$

where$y$istheone-hottargetand ε=0.1.TheAdamWoptimiser[20]wasusedwiththesettingsinTableIII.The checkpoint with the highest PlantDoc validation accuracy was kept.

TABLEIII
FINE-TUNINGHYPERPARAMETERS

Parameter	Value
Initialmodel	ViT-B/16,PlantVillagecheckpoint
Optimiser	AdamW,weightdecay0.05
Learningrate/schedule	3×10^{-5},cosine,6% warm-up

Batchsize/epochs	32/8
Loss	Cross-entropy,labelsmoothing0.1
Precision/gradientclip	FP16mixedprecision/norm1.0
Hardware	NVIDIATeslaT4(16GB),GoogleColab

F. Quantisation and Deployment

Theselectedmodelwasexportedto ONNX(opset17)andquantisedwithdynamicpost-trainingquantisation [8]: weights are stored as unsigned 8-bit integers and activations are quantised at run time. This reduces the file to 87 MB and allows inference with ONNX Runtime on the CPU, so PyTorch is not needed in production. The complete service uses about 230 MB of memory, which fits the 512 MB limit of free cloud plans.

G. DecisionLogic,SeverityandHealthScore

The response status is *no_leaf* if segmentation fails, *uncertain* if the top-1 probability is below 0.55, and otherwise *healthy* or *diseased* according to the predicted class. For diseased leaves, severity is graded as *early* (S<5 %), *moderate* (5 % ≤ S < 20 %) or *severe* (S ≥ 20 %). A health score $H \in [5, 95]$ combines the urgency of the disease, taken from the knowledge base, with the measured severity:

$$H = \mathrm{clip}(100 - P_u - 1.5S, 5, 95), \quad (6)$$

where P_u is 25, 12 or 5 for high, medium or low urgency, respectively. A healthy leaf scores 100, or 90 if more than 10 % of its area shows discolouration.

H. KnowledgeBaseandAssistant

A curated JSON knowledge base contains an entry for each of the 38 classes with symptoms, cause, chemical and organic treatment, prevention measures and urgency, together with crop information and Hindi crop names; a short Wikipedia summary of the crop is added at run time. The assistant forwards the user's question, the conversation history and the current diagnosis to Google Gemini [17] under a system prompt restricted to farming topics, and can reply in nine languages (English, Hindi, Marathi, Tamil, Telugu, Kannada, Bengali, Gujarati and Punjabi). When no API key is configured or the network is unavailable, it answers from the local knowledge base instead.

I. Justification of Technology Choices

A Vision Transformer was selected because self-attention relates distant lesions across the whole leaf and transfers well from pre-training [7], [15]. ONNX Runtime with int8 weights was chosen over PyTorch to meet the memory limit of free hosting. Classical colour segmentation was preferred for localisation because pixel-levellesion annotations, whichalearnedsegmentationmodelwouldrequire,arenotavailableforthe38classes.FastAPI providesasynchronousrequesthandlingand automaticAPIdocumentation, whileaframework-freePWAkeepsthe client small, installable and usable on low-end Android phones.

IV. IMPLEMENTATION

A. Modules

The backend consists of seven Python modules: vision.py (segmentation, Algorithm 1), classifier.py (ONNX inference),diagnosis.py(pipeline,status, severityand healthscore), knowledge.py(knowledge base and Wikipedia lookup), assistant.py (Gemini and offline answers), db.py (SQLite history, capped at 500 scans to bound disk use) and main.py (REST endpoints for diagnosis, liveframes, scan history,thedisease library and chat). The frontend is organised into modules for the AR scanner and overlay renderer, the result sheet with read-aloud via the Web Speech API, the history view with a health-trend chart, the searchable disease library, and the chat assistant with voice input. A service worker caches the application shell so that the interface opens offline.

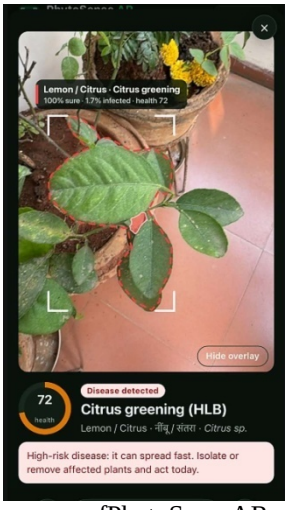


Fig.3.ARlivescannerofPhytoSenseARonasmartphone

B. Development Environment

The backend was developed in Python with FastAPI and Uvicorn, OpenCV, NumPy, Pillow and ONNX Runtime. Fine-tuning used PyTorch,Hugging Face Transformers and torchvision in a Google Colab notebookona free Tesla T4 GPU. The frontend uses HTML5, CSS and JavaScript with the MediaDevices camera API and the Canvas 2D API. Automated tests written with pytest cover segmentation, the knowledge base, the API, the history limit and the assistant; all eight tests pass with the fine-tuned model. The service is containerised with Docker and deployed on the Render free plan, which provides the HTTPS connection that browsers require for camera access.

TABLEIV
HARDWAREANDSOFTWAREREQUIREMENTS

Component	Requirement
ServerCPU/RAM	Anyx86-64CPU,≥512MBRAM(≈230MBused)
ServerGPU	Not required
Storage	≈100 MB(model87MB)+scanphotos
Client	SmartphoneorPCwithcameraandmodernbrowser
Serversoftware	Python≥3.10,FastAPI,ONNXRuntime,OpenCV
Training(one-off)	GoogleColab,NVIDIAT4GPU,PyTorch
Connectivity	HTTPS;InternetonlyforWikipediaandGemini

V. RESULTS AND DISCUSSION

A. Evaluation Protocol

Top-1 accuracy was measured on two test sets that were never used for training or model selection: the official PlantDoc test split (236 field images, 27 classes) and the 760 held-out PlantVillage images. Both sets were preprocessed exactly as in the application. The laboratory figure should be interpreted as an upper bound, because the starting checkpoint was trained by a third party on PlantVillage and may already have seen some of these images.

B. Training Behaviour

Fig. 4 shows that the training loss decreased from 1.355 to 0.711 over eight epochs. PlantDoc validation accuracy reached 76.1 % after the first epoch and peaked at 79.5 % in epoch 7, whose weights were retained. Each epoch took about 185 s on the T4 GPU, so fine-tuning completed in approximately 25 minutes.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

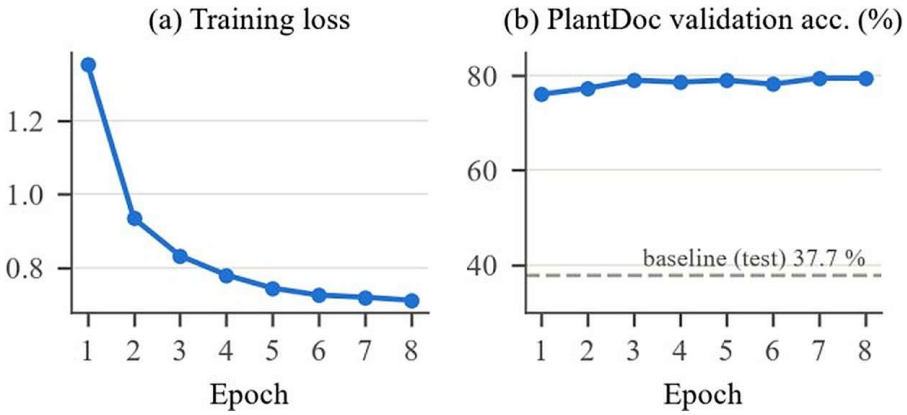


Fig.4.Fine-tuningprogress:(a)trainingloss(withlabelsmoothing)and(b)accuracyonthePlantDocvalidationsplit.The dashed line marks the original model's accuracy on the PlantDoc test set

C. Classification Accuracy

TableVandFig.5comparetheoriginalandfine-tunedmodels.Fieldaccuracyapproximatelydoubled,from37.7 % to 75.4 % for the FP32 model and 76.7 % for the deployed int8 model, while laboratory accuracy stayed at or near 100 %. The small advantage of the int8 model corresponds to three images and lies within sampling variation; with 236 test images, the 95 % confidence interval of the int8 result is about ±5.4 percentage points. Quantisation therefore did not measurably reduce accuracy.

TABLEV
TOP-1ACCURACYBEFOREANDAFTERFINE-TUNING

Model	PlantDoctest(field)	PlantVillageheld-out
Original(PlantVillageonly)	37.7%	100.0%
Fine-tuned,FP32	75.4%	100.0%
Fine-tuned,int8ONNX(deployed)	76.7 %	99.9 %

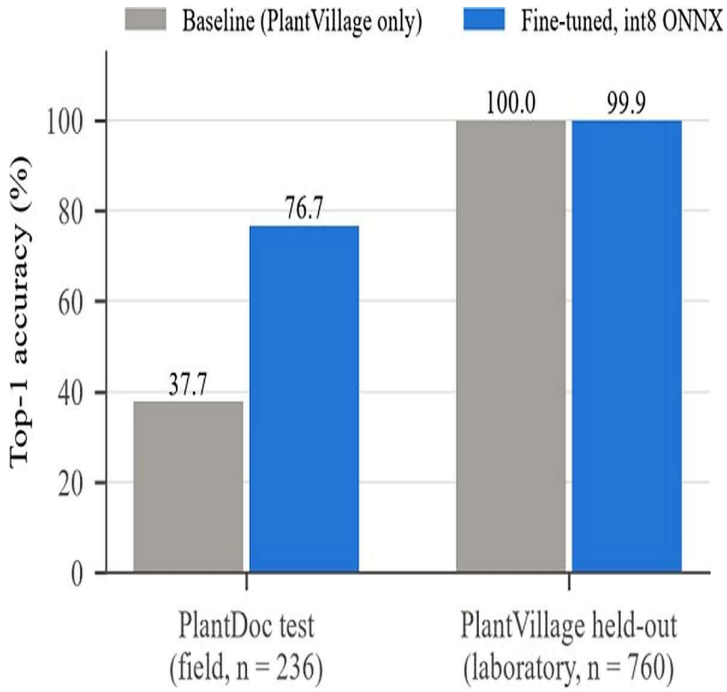


Fig.5.Top-1accuracyoftheoriginalandthedeployedfine-tunedmodelonfield(PlantDoc)andlaboratory(PlantVillage)test images.

D. Per-ClassAnalysis

Fig. 6 reports per-class field accuracy. Healthy leaves of most crops, squash powdery mildew, apple scab, corn common rust and grape black rot are recognised with 88–100 % accuracy, because their symptoms or textures are visually distinctive. The weakest classes are corn gray leaf spot (25 %), potato early blight (38 %) and tomato bacterial spot (44 %). Their errors mostly fall into visually similar classes of the same crop, such as northern leaf blight versus gray leaf spot, early versus late blight, and bacterial versus Septoria leaf spot, whose small necrotic lesionsarehard toseparateinlow-resolutionfield photos.Eachclasshasonly4–12testimages, soper-classfigures carry large uncertainty.

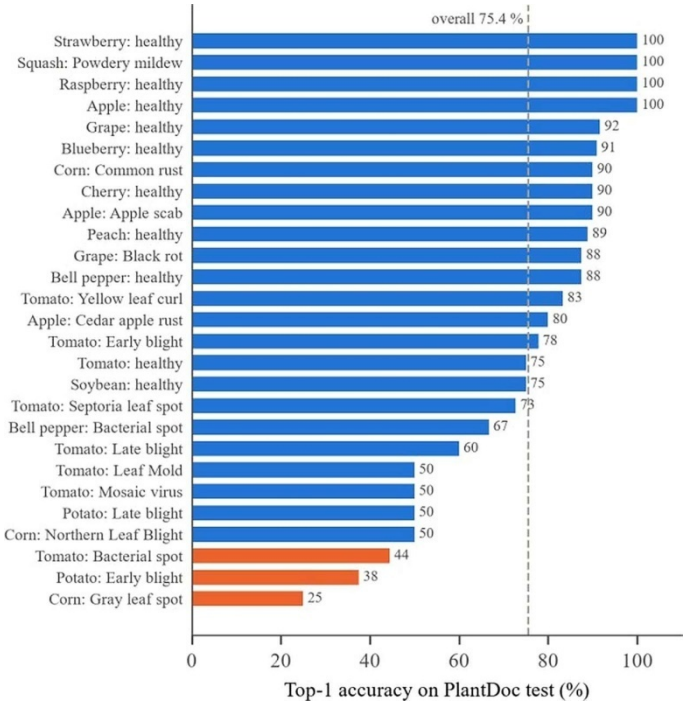


Fig.6.Per-classtop-1accuracy ofthefine-tunedmodelonthePlantDoctestset.Orangebarsmarkclassesbelow50%;the dashed line marks the overall accuracy

E. RuntimePerformance

Table VI lists the runtime characteristics measured for the deployed configuration. On a laptop CPU, segmentationandclassificationtogethertakeabout190–200msperimage.Aseparatecheckon40randomlydrawn PlantVillage test images from a public mirror gave 97.5 % accuracy with a mean confidence of 90.5 %. The lower confidence compared with the original model (98.3 % mean on a similar check) is an expected effect of label smoothing and makes the 0.55 confidence threshold more selective.

TABLEVI
RUNTIMECHARACTERISTICSOFTHEDEPLOYEDSYSTEM

Metric	Value
Modelsize(int8ONNX)	87MB
Servermemoryuse	≈230MB
Timeperimage,laptopCPU	≈190–200ms
LiveAROverlayrate,laptop CPU	≈2–3updates/s

F. ComparisonwithExistingSystems

Table VII compares PhytoSense AR with representative approaches from Section II. Direct numerical comparison is avoided because the cited works use different datasets, class sets and test splits. The comparison shows that the proposed system is, among these works, the only one that combines field-adapted classification, visual localisation, severity estimation, treatment advice and zero-cost CPU deployment.

TABLEVII
FEATURECOMPARISONWITHEXISTINGAPPROACHES

Feature	[3],[9],[10]	[6]	[14]	[15],[16]	Ours
Field-imagetraining	No	Yes	Yes	Partly	Yes
Lesionlocalisation/AR	No	Boxes	Boxes	No	Yes
Severityestimate	No	No	No	No	Yes
Treatmentadvice	No	No	No	No	Yes
Multilingualassistant	No	No	No	No	Yes
RunsonCPU,no cost	—	—	Phone	Partly	Yes

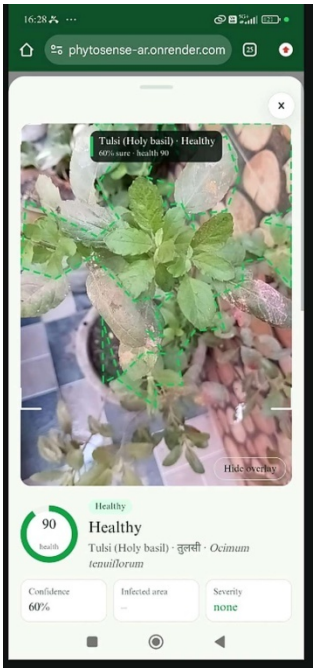


Fig.7.ResultandhistoryscreensofPhytoSense AR

G. Discussion and Limitations

Theresultsconfirmthemainhypothesisof thework:amodelthatisalmostperfectonlaboratoryimagescanfail on field images, and modest field data combined with realistic augmentation largely closes this gap. Several limitations remain. The classifier is closed-set: a plant outside the 14 supported cropsis still assigned one of the 38 labels, which the confidence threshold mitigates onlypartly. Ten classes, including citrusgreening, grape Esca and tomato target spot, have no field training images and rely on laboratory data alone. The colour-based segmentation performs well for distinct spots, blights, rusts and mildews but is weaker for diffuse symptoms such as mosaic mottling or mite damage, and for scenes with many overlapping leaves; its severity values have not yet been validated against expert annotations. Finally, the AR layer is a two-dimensional overlay on the video rather than world-anchored 3-D AR, and live mode requires a network connection to the server.

VI. CONCLUSION AND FUTURESCOPE

ThispaperpresentedPhytoSenseAR,afreeAR-basedsystemforplantdiseasediagnosisthatcombinestraining-free leaf and lesion segmentation, a field-adapted Vision Transformer, severity and health scoring, a treatment knowledge base and a multilingual assistant. Fine-tuning on PlantVillage and PlantDoc with field-style augmentation raised top-1 accuracy on unseen field photographs from 37.7 % to 76.7 % while keeping laboratory accuracy at 99.9 %, and int8 quantisation allowed the whole system to run on a CPU in about 200 ms per image within a free cloud instance. These results show that practical, explainable plant diagnosis does not require expensive hardware or paid services.

©IJRASET: All Rights are Reserved

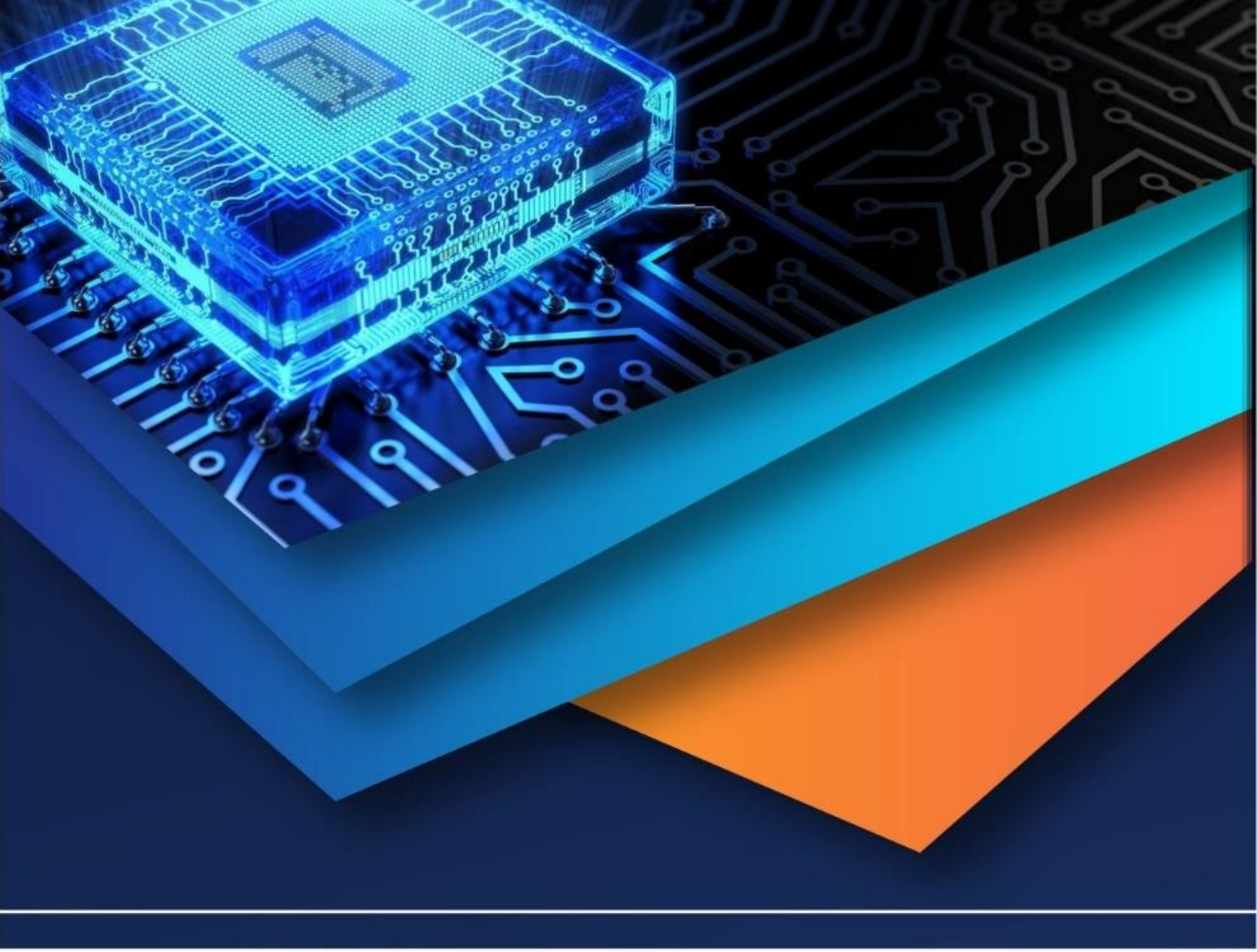
Futureworkwilladdresstheidentifiedlimitations:(i)trainingalightweightinstance-segmentationnetworksuch as YOLO-seg on PlantDoc annotations to localise lesions on multi-leaf scenes; (ii) running inference on the phone with ONNX Runtime Web to remove thenetworkdependency oflive mode;(iii) addingan open-setrejectionstage or crop selection to handleunsupportedplants; (iv) collecting field imagesof crops important in India, such asrice, wheat and cotton; and (v) validating the severity estimate and the usability of the application in a field study with farmers.

VII. ACKNOWLEDGMENT

The authors express sincere gratitude to Dr. Puja Gupta, Assistant Professor, Department of Information Technology, Shri G. S. Institute of Technology and Science, Indore, for her valuable guidance, constructive feedback and continuous encouragement throughout this project. The authors also thank the Department of Information Technology for providing the necessary facilities, and the creators of the PlantVillage and PlantDoc datasets for making their data openly available.

REFERENCES

[1] S. Savary, L. Willocquet, S. J. Pethybridge, P. Esker, N. McRoberts, and A. Nelson, "The global burden of pathogens and pests on major food crops," Nature Ecology & Evolution, vol. 3, no. 3, pp. 430–439, 2019.
[2] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv preprint arXiv:1511.08060, 2015.
[3] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Frontiers in Plant Science, vol. 7, art. no. 1419, 2016.
[4] K.P.Ferentinos,"Deeplearningmodelsforplantdiseasedetectionanddiagnosis,"ComputersandElectronicsin Agriculture, vol. 145, pp. 311–318, 2018.
[5] J.G.A.Barbedo,"Factorsinfluencingtheuseofdeeplearningforplantdiseaserecognition,"BiosystemsEngineering,vol. 172, pp. 84–91, 2018.
[6] D. Singh, N. Jain,P. Jain,P. Kayal, S. Kumawat, andN. Batra,"PlantDoc: Adatasetforvisualplantdisease detection," in Proc. 7th ACM IKDD CoDS and 25th COMAD, Hyderabad, India, 2020, pp. 249–253.
[7] A. Dosovitskiyet al., "An image is worth 16x16 words: Transformers for image recognition at scale," in Proc. Int. Conf. Learning Representations (ICLR), 2021.
[8] B.Jacobet al.,"Quantizationandtrainingofneural networksforefficientinteger-arithmetic-onlyinference,"inProc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 2704–2713.
[9] E.C.Too,L.Yujian,S.Njuki,andL.Yingchun,"Acomparativestudyoffine-tuningdeeplearningmodelsforplant disease identification," Computers and Electronics in Agriculture, vol. 161, pp. 272–279, 2019.
[10] Ü.Atila,M.Uçar,K.Akyol,andE.Uçar,"PlantleafdiseaseclassificationusingEfficientNetdeeplearningmodel," Ecological Informatics, vol. 61, art. no. 101182, 2021.
[11] J.G.A.Barbedo,"Plantdiseaseidentificationfromindividuallesionsandspotsusingdeeplearning,"Biosystems Engineering, vol. 180, pp. 96–107, 2019.
[12] L. Li, S. Zhang, and B. Wang, "Plant disease detection and classification by deep learning—A review," IEEE Access, vol. 9, pp. 56683–56698, 2021.
[13] A. Ahmad, D. Saraswat, and A. El Gamal, "A survey on using deep learningtechniques for plant disease diagnosis andrecommendations for development of appropriate tools," Smart Agricultural Technology, vol. 3, art. no. 100083, 2023.
[14] A. Ramcharanetal.,"Amobile-based deeplearning modelfor cassavadisease diagnosis," Frontiers in PlantScience, vol. 10, art. no. 272, 2019.
[15] Y.Borhani, J.Khoramdel,and E.Najafi,"A deeplearning basedapproachforautomatedplantdiseaseclassification using vision transformer," Scientific Reports, vol. 12, art. no. 11554, 2022.
[16] P. S. Thakur, P. Khanna, T. Sheorey, and A. Ojha, "Explainable vision transformer enabled convolutional neural network for plant disease identification: PlantXViT," arXiv preprint arXiv:2207.07919, 2022.
[17] GeminiTeam,Google,"Gemini:Afamilyofhighlycapablemultimodalmodels,"arXivpreprintarXiv:2312.11805,2023.
[18] D.M.Woebbecke,G.E.Meyer,K.VonBargen,andD.A.Mortensen,"Colorindicesforweedidentificationunder various soil, residue, and lighting conditions," Transactions of the ASAE, vol. 38, no. 1, pp. 259–269, 1995.
[19] C. Szegedy, V.Vanhoucke,S.Ioffe,J.Shlens,and Z. Wojna,"Rethinking theInceptionarchitectureforcomputervision," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2818–2826.
[20] I.LoshchilovandF.Hutter,"Decoupledweightdecayregularization,"inProc.Int.Conf.LearningRepresentations (ICLR),2019.
[21] Gupta, P., Sharma, V. and Varma, S., 2021. People detection and counting using YOLOv3 and SSD models. Materials Today: Proceedings.
[22] Gupta, P., Shukla, M., Arya, N., Singh, U. and Mishra, K., 2022. Let the Blind See: An AIIoT-Based Device for Real-Time Object Recognition with the Voice Conversion. In Machine Learning for Critical Internet of Medical Things (pp. 177-198). Springer, Cham
[23] Gupta, P. and Kulkarni, N., 2013. An introduction of soft computing approach over hard computing. International Journal of Latest Trends in Engineering and Technology (IJLTET), 3(1), pp.254-258
[24] Gupta, P., Sharma, V. and Varma, S., 2022. A novel algorithm for mask detection and recognizing actions of human. Expert Systems with Applications, 198, p.116823.
[25] Gupta, P., Kulkarni, A. and Sarda, A., 2013. An embedded health care supervisory systems. International Journal of Latest Trends in Engineering and Technology (IJLTET), 3, pp.379-386



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)