Dr. B R Ambedkar National Institute of Technology

Minor Project Presentation (IT Department)

Title – Crop Disease Detection

Domain – Smart Agriculture

PRESENTED BY -

ISHAAN (23124042)

NAMRATHA (23124066)

KHUSHI MITTAL (23106050)

PROJECT GUIDE -

DR. JASPAL KAUR SAINI

ASSISTANT PROFESSOR

DEPARTMENT OF IT

CONTENT

INTRODUCTION

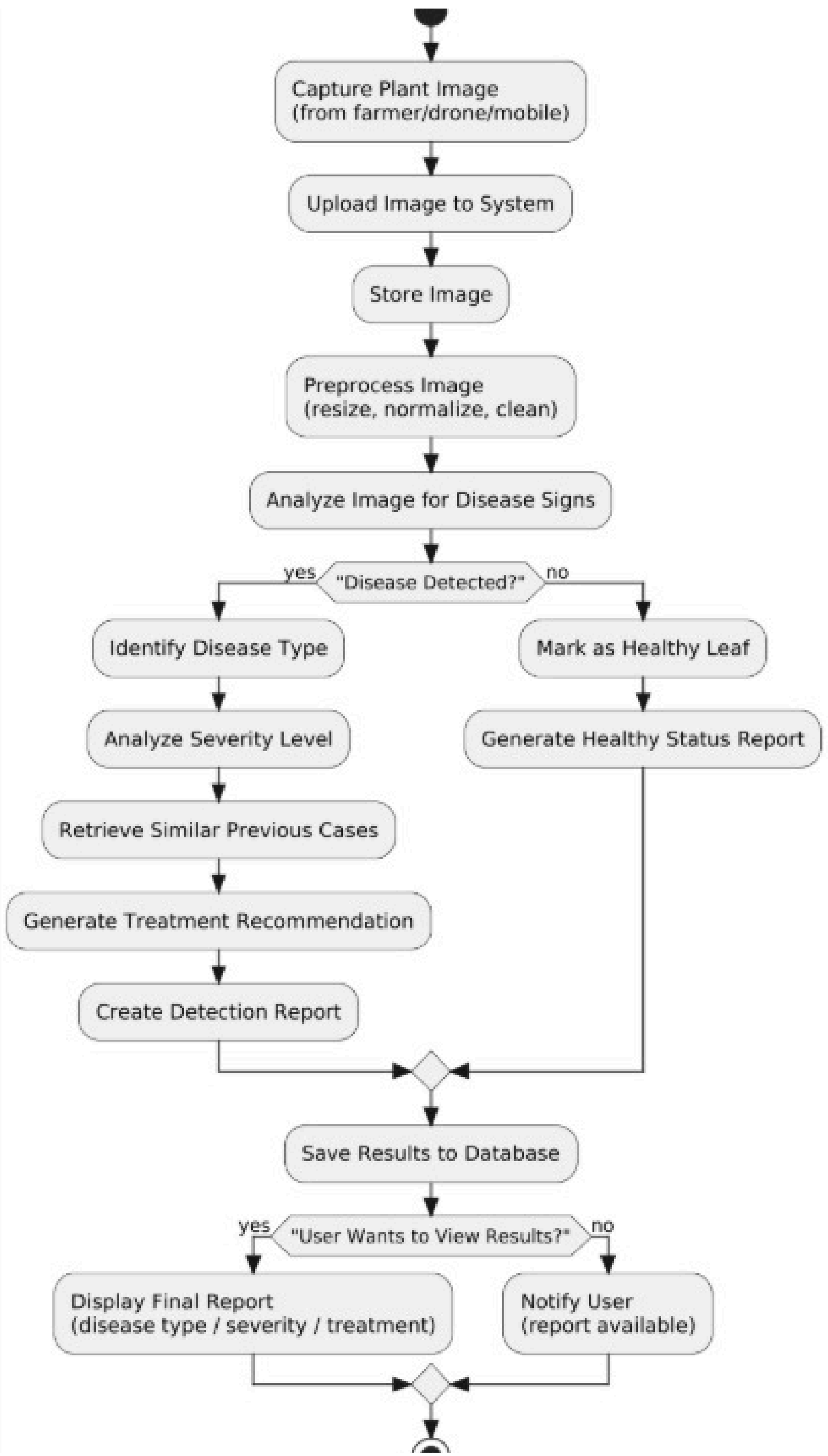
- Crop diseases reduce yield and quality, causing major economic losses.
- Manual monitoring is :
  - Labor-intensive
  - Time-consuming
  - Requires expert knowledge
  - Often leads to delayed detection
- AI and Deep Learning enable automated crop disease detection using image analysis.
- But current models are :
  - Computationally heavy
  - Unsuitable for low-cost or mobile devices used in rural areas
- Hence, there is a need for a lightweight and accurate system capable of detecting crop diseases using images.

Proposed Smart Agriculture Solution

- Lightweight deep learning models optimized for edge deployment on mobile or low-power devices.
- Real-time image processing from drones or smartphone to monitor crops.
- Automatic disease detection and classification for timely intervention.
- Treatment recommendations or alerts to guide farmers in managing crop health.

Motivation:

Crop diseases reduce yield and threaten food security. Mannual detection is slow and inefficient, while lightweight deep learning models enable early detection and promote sustainable farming.



CONVOLUTIONAL NEURAL NETWORKS

Neural Networks : **A computational model inspired by the way the human brain processes information.**

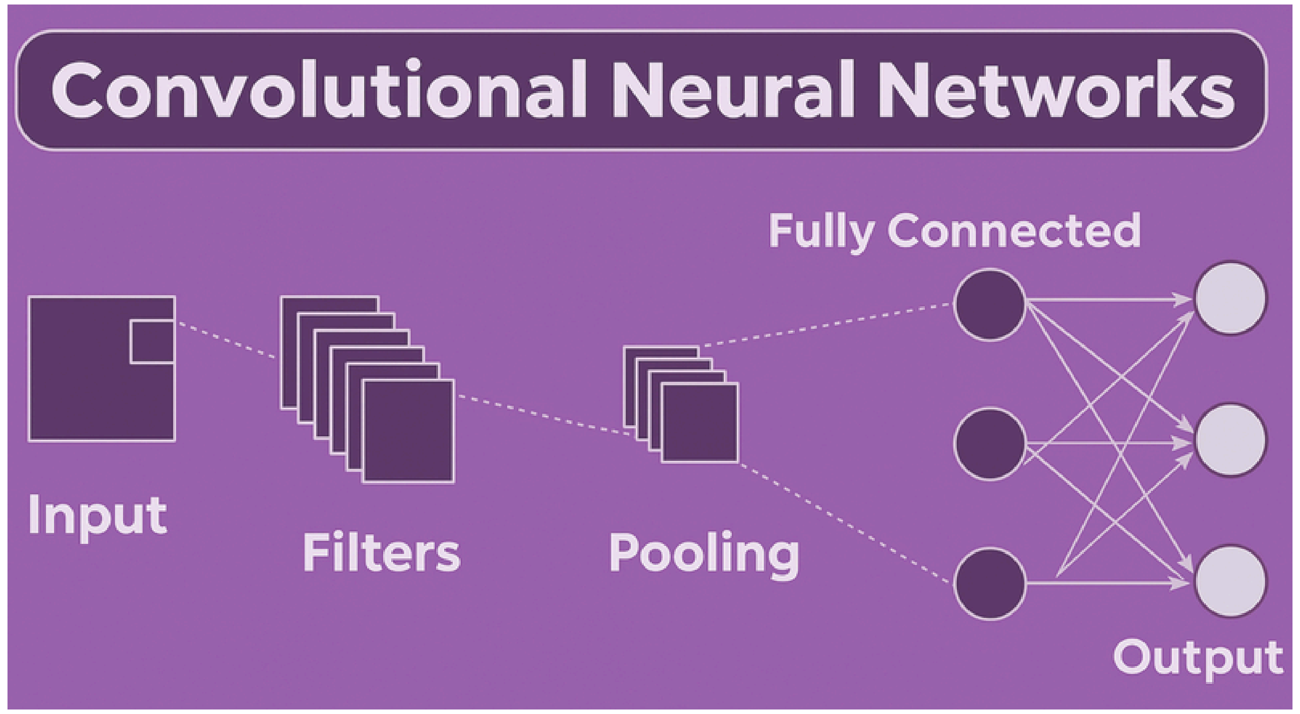
- It consists of layers of interconnected nodes (neurons) where each neuron receives input processes it, and passes an output to the next layer.
- Learns from data over time to recognize patterns, make predictions, and classify images or objects.

Convolutional Neural Networks (CNNs)

A specialized type of neural network designed to recognize grid-like patterns such as images. It uses filters (kernels) that slide over the image to detect features and textures.

Key Layers:

- **Convolution Layer:** Extracts important features using filters.
- **Pooling Layer:** Reduces dimensions , retains key info.
- **Fully Connected Layer:** Combines learned features for final classification.



DEEP LEARNING MODELS

Deep Learning Models are advanced types of machine learning models that use artificial neural networks with many layers to automatically learn patterns and features from large amounts of data.

ShuffleNet

ShuffleNet is designed for high-speed and low-power applications, combining group convolutions and a unique channel shuffle operation to reduce computation cost.

Balances accuracy and efficiency, ideal for real-time image recognition.

SqueezeNet

Lightweight CNN achie ving AlexNet-level accuracy with 50× fewer parameters, ideal for low-resource devices.

Uses Fire modules to efficiently squeeze and expand feature maps, reducing size without losing accuracy.

EfficientNet

EfficientNet is a deep learning model that balances depth, width, and resolution using compound scaling, achieving high accuracy with fewer parameters and computations.

Delivers high accuracy with fewer parameters and computations.

Yolo

YOLO is a real-time object detection model that detects and classifies multiple objects in a single pass through the neural network, enabling fast and accurate identification of objects within images or video streams.

It offers fast and accurate detection, making it ideal for real-time applications like surveillance and autonomous systems.

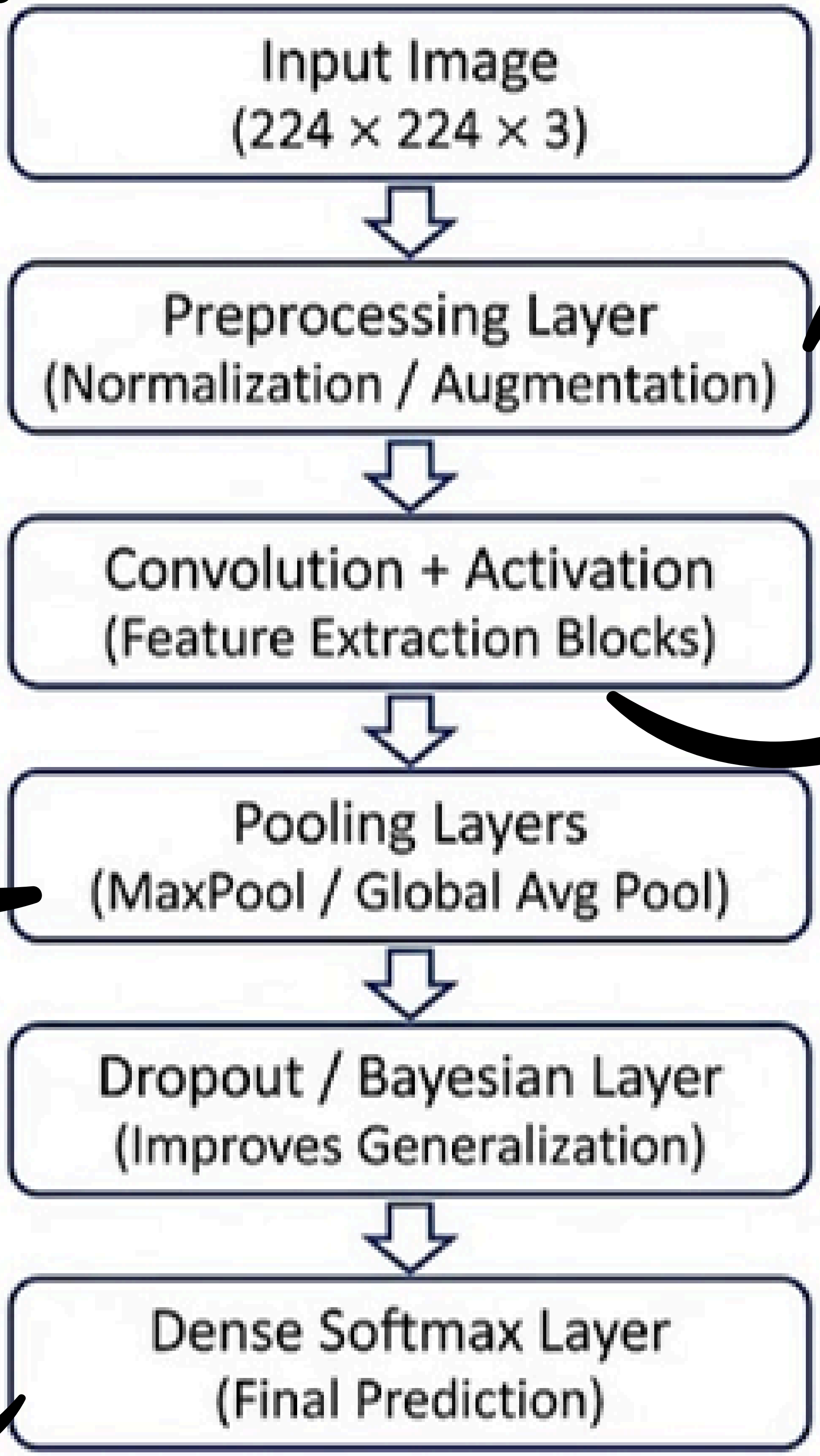
MobileNet

MobileNet is a lightweight convolutional neural network specifically designed for mobile and embedded devices. It uses depthwise separable convolutions to significantly reduce computational cost.

It provides fast and efficient image recognition, making it ideal for real-time applications on mobile and embedded devices, where computational resources and power are limited.

Detailed Overview

BUILDING THE CNN MODELS



Standard Preprocessing: Normalization , Resizing and Data Augmentation

- Reduces spatial dimensions
- Keeps most important features , improves computational effeciency
- max pooling , average pooling , gap pooling

It has the following hiereical stages -
Low-Level Features - Edges , Corners , Texture patterns
- Mid-Level Features - Spots patches , Leaf veins / shapes
- High-Level Features - Disease-specific patterns , Shape irregularities , Localized discoloration

- Fully Connected (Dense) Layer
- Softmax activation for multi-class classification
- Produces final probability distribution over diseases

All models in this project follow a typical CNN pipeline that extracts visual patterns from images and classifies them into crop disease categories.
Despite differences in architecture (EfficientNet, MobileNet, SqueezeNet), each model uses the same core steps as demomstarted in the previous slide. The common traning configuration/parameters for the mdoels are given as follows:

SNO.	PARAMETER	VALUE
1.	Epochs	2–20 depending on experiment
2.	Batch Size	32
3.	Metrics	Accuracy, Precision, Recall, F1-score
4.	Callbacks	ModelCheckpoint + EarlyStopping

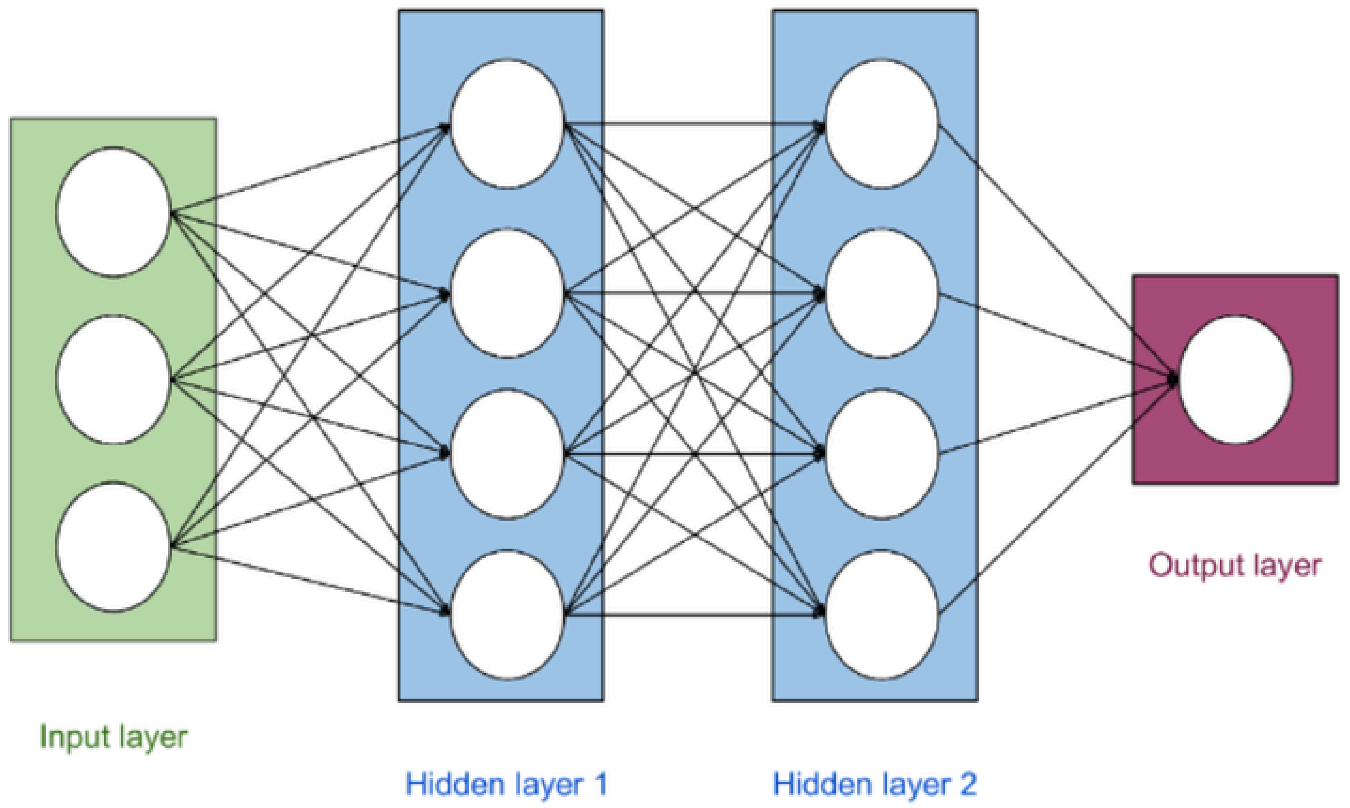
The Bayesian OPTIMIZATION

A Bayesian layer is a neural network layer that models uncertainty in predictions instead of producing fixed, deterministic outputs.

- **Bayesian layers introduce controlled randomness (like MC Dropout), which forces the model to Learn smoother decision boundaries , Not memorize the training data , Focus on important patterns instead of noise**

- **Adding Bayesian layers in different parts of a CNN affects performance differently:**

1.	Inside Early Convolution Layers	Adding dropout in very early layers affects basic feature extraction
2.	In Middle Convolution Blocks	Introduces randomness when mid-level abstract features are forming
3.	After Feature Extraction	Bayesian dropout applied on high-level features
4.	After Final Dense Layer	Adding randomness before final prediction , not meaningfull



THE FINAL RESULT

1) EFFECIENT NET B0 (CODE)

DATASET USED – PLANT VILLAGE

- Training: 18,575 images
- Validation: 2,063 images
- Total Classes: 15 crop disease categories

MODEL DESCRIPTION –

- Total params: 4,068,786
- Trainable params: 19,215
- Non-trainable params: 4,049,571
- Effective Layers: 1 Input + 1 Base Model + GAP + Dropout + Dense

BAYESIAN LAYER –

- Approximation: MC Dropout
- Applied after GlobalAveragePooling2D
- Dropout rate: 0.5
- During inference, Dropout(active=True) gives uncertainty

RESULT –

- Improvement: +13.5% accuracy
- Observation: Faster convergence, reduced overfitting, smoother loss curve.
- Conclusion: Best performance comes from adding Bayesian regularization right before the final Dense classifier.

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 224, 224, 3)	0
efficientnetb0 (Functional)	(None, 7, 7, 1280)	4,049,571
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0
dropout (Dropout)	(None, 1280)	0
dense (Dense)	(None, 15)	19,215

BEFORE

```
65/65 ━━━━━━━━━━━━━━━━━━━━ 2s 31ms/step - accuracy: 0.7785 - loss: 0.9632
Validation Accuracy: 0.7945
```

AFTER

```
65/65 ━━━━━━━━━━━━━━━━━━━━ 1s 20ms/step - accuracy: 0.9266 - loss: 0.2829
Quick Validation Accuracy: 0.9292292594909668
```

Metric	Before	After(Bayesian)
ACCURACY	79.45%	92.92%
PRECISION , RECALL	77.25% , 79.39%	85.86% , 85.55%
F1 – Score	77.03%	84.8%

2) SQUEEZE NET

DATASET USED –NEW PLANT DISEASES DATASET (AUGMENTED)

- Training Images: ~70,000
- Validation Images: ~18,000
- Total Classes: 38 plant disease + healthy leaf categorie

MODEL DESCRIPTION –

- Base Architecture: SqueezeNet 1.1 (pre-trained on ImageNet)
- Total Parameters: ~1.2 million
- Trainable Parameters: ~735,000
- Uses fire modules to achieve AlexNet-level accuracy with 50× fewer parameters, making it fast and lightweight.

BAYESIAN LAYER –

- Technique Used: Bayesian MC-Dropout
- Location Applied: Dropout layer added before the final Conv classifier
- Dropout Rate: 0.5
- Purpose: Introduces uncertainty and regularization to reduce overfitting

RESULT –

- Before Bayesian Accuracy: ~87%
- After Bayesian Accuracy: ~97%
- Improvement: +10% accuracy boost

BEFORE

```
Epoch [1/5]: 100%|
Validation Accuracy after epoch 1: 5.28%
Epoch [2/5]: 100%|
Validation Accuracy after epoch 2: 71.86%
Epoch [3/5]: 100%|
Validation Accuracy after epoch 3: 83.64%
Epoch [4/5]: 100%|
Validation Accuracy after epoch 4: 88.58%
Epoch [5/5]: 100%|
Validation Accuracy after epoch 5: 87.18%
✅ Model saved as squeezenet_crop_disease.pth
khushimittal@KHUSHIs-MacBook-Air-2 SqueezeNet %
```

AFTER

```
Epoch [1/5]: 100%|
Validation Accuracy after Epoch 1: 91.21%
Epoch [2/5]: 100%|
Validation Accuracy after Epoch 2: 91.92%
Epoch [3/5]: 100%|
Validation Accuracy after Epoch 3: 95.52%
Epoch [4/5]: 100%|
Validation Accuracy after Epoch 4: 93.85%
Epoch [5/5]: 100%|
Validation Accuracy after Epoch 5: 97.06%

Model saved as fine_tuned_bayesian_mcdropout.pt
khushimittal@KHUSHIs-MacBook-Air-2 SqueezeNet %
```

Metric	Before	After(Bayesian)
ACCURACY	87%	97%
PRECISION , RECALL	88.74% , 87.10%	97.14% , 97.03%
F1 – Score	87.12%	97.01%

3) MOBILENETV2 (CODE)

DATASET USED – TOMATO LEAF DISEASE

- Training Images: 8,000
- Validation Images: 2,000
- Total Classes: 10 tomato disease categories

MODEL DESCRIPTION –

- Total parameters: ~3.5 million
- Trainable parameters: ~135,000
- Base Architecture: MobileNetV2 (pre-trained on ImageNet)
- Uses depthwise separable convolutions to reduce computation

BAYESIAN LAYER –

- Applied Dropout layers before the final Dense classifier
- Dropout rate: 0.5
- Multiple forward passes (20 samples) are averaged to compute:
  1) Standard deviation (uncertainty)
  2) Mean prediction (confidence)

RESULT –

- Improvement: -7.3 % accuracy
- Observation: Enhanced generalization with reduced overfitting, better-calibrated predictions, and increased robustness
- Conclusion: Best practical performance is achieved by introducing Bayesian MC-Dropout before the final Dense classifier

Layer (type)	Output Shape	Param #
input_1	(None, 224, 224, 3)	
mobilenetv2_1 (Functional)	(None, 7, 7, 1280)	0
global_average_ (GlobalAveragePooling2d)	(None, 1280	0
dropout	(None, 1280	0
dense	(None, 15	19,335

BEFORE

```
💾 Model saved as mobilenet_leaf_model.h5
63/63 ━━━━━━━━━━━━━━━━━━━━ 58s 912ms/step - accuracy: 0.8540 - loss: 0.4348

✅ Validation Accuracy: 85.40%
```

AFTER

```
Validation Accuracy: 78.10%
```

Metric	Before	After(Bayesian)
ACCURACY	85.40 %	78.10 %
PRECISION , RECALL	88%, 85%	80%, 78%
F1 - Score	86%	79%

Year	Name	Data Set	Model / Library	Findings	Accuracy	Limitations
1) 2019	**YOLO Nano: A Compact You Only Look Once CNN for Object Detection**	• PASCAL VOC datasets, VOC2007/2012 • Natural images for 20 different objects. • Performance compared to Tiny YOLOv2 and v3.	• YOLO Nano • Created using a human-machine collaborative design strategy. • Unique residual projection-expansion-projection (PEP) macroarchitectures and lightweight fully-connected attention (FCA)	• Solves high computational and memory needs by balancing accuracy, size, and computational complexity. • Generative synthesis used to create a highly tailored and efficient network	• YOLO Nano achieved a 69.1% mAP on the VOC 2007 dataset, outperforming Tiny YOLOv2 and v3. • Demonstrated high inference speeds, power efficiency on an embedded module	• Existing models are often too slow and large for embedded devices • Lighter versions like Tiny YOLO compromise on accuracy to reduce size
2) 2024	**Tiny Yolo Networks for Disease Detection in Paddy Agronomy**	• Research team constructed their own rice leaf disease dataset • 3 classes: Bacterial leaf blight, Rice blast, and Brown spot identification. • Final dataset size - 15,456 images.	• Modified version of Tiny YOLOv4 • UAV T-YOLO-Rice for detecting rice leaf diseases • Spatial Pyramid Pooling (SPP), a Convolutional Block Attention Module, Sand Clock Feature Extraction Module (SCFEM).	• Detecting small, diseased areas on rice leaves from UAVs • Enhanced model allows for higher accuracy and better detection of small objects	• The model achieved a testing mAP of 86.97%, which is the highest among the compared models • Outperformed the base Tiny YOLOv4 model and also YOLOv7, YOLOv2	• Difficulty of detecting small infection points using UAV cameras • Model has low accuracy despite its real-time detection capability

Year	Name	Data Set	Model / Library	Findings	Accuracy	Limitations
3) 2020	**EfficientNet: Rethinking Model Scaling for CNNs**	• Models primarily evaluated on the ImageNet dataset • Tested on CIFAR-100 (91.7%), Flowers (98.8%), and 3 other transfer learning datasets • Models fine-tuned on new datasets after being pre-trained on ImageNet	• EfficientNet, a family of CNNs • EfficientNet-B0, uses the mobile inverted bottleneck MBConv as its main building block and incorporates squeeze-and-excitation optimization	• Studies model scaling, for improving ConvNet accuracy • Balance and scale all 3 dimensions (depth, width, and resolution) uniformly using a compound coefficient	• Achieved a state-of-the-art 84.4% top-1 accuracy on ImageNet & 97.1% top-5 accuracy on ImageNet. • Outperformed other ConvNets on ImageNet with fewer parameters	• Arbitrary scaling requires tedious manual tuning • Effectiveness of model scaling is heavily dependent on baseline network used.
4) 2025	**Optimized classification using EfficientNet-LITE & KE-SVM in diverse environments**	• One dataset was a public one from a Kaggle source of an uncontrolled environment with 3,076 images from Indonesia • 2nd dataset is PlantVillage Dataset (Potato Species), which has 2,152 images from a controlled laboratory setting	• EfficientNet-LITE with KE-SVM Optimization was developed to classify potato leaf diseases • Model integrates Channel Attention and 1-D Local Binary Pattern features • Implemented using libraries like Keras and OpenCV.	• Proposes a hybrid approach combining a lightweight DL model for feature extraction and an optimized machine learning classifier • Create efficient model for use on edge devices	Overall accuracy : Uncontrolled environment dataset -87.82% Controlled dataset - 99.54% • Performance on uncontrolled data shows significant improvement over other models	• Dataset lacks diversity for real-world applications • Failed to effectively combine lightweight models with robust optimization techniques

Year	Name	Data Set	Model / Library	Findings	Accuracy	Limitations
5) 2023	**An Improved MobileNet for Disease Detection on Tomato Leaves**	• PlantVillage Tomato Leaf Dataset (Kaggle). • 2,064 images → 4 classes: Early Blight, Septoria, Yellow Curl, Healthy. • Images resized • Resized + augmented (rotation, zoom, flip, etc.) • Dataset expanded to 4,128 images.	• Improved MobileNet - a lightweight CNN. • Transfer learning and fine-tuning (Keras/TensorFlow). • Depthwise separable convolutions to reduce parameters.	• Automates detection of tomato leaf diseases. • Outperforms VGG16, VGG19, DenseNet201, and Xception models. • Achieves higher accuracy and faster training speed.	Overall accuracy : 97.17% (multi-class) Per-class : • Early Blight -98.57% • Yellow Curl -99.52% • Septoria - 98.18% • Healthy - 99.54% F1-score: 0.9442	• Only 4 disease classes. • Lacks real-world diversity. • Model not tested in real-time field conditions. • Performance may vary with unseen diseases.
6) 2023	**Improved MobileNetV2 crop disease identification model for intelligent agriculture**	• PlantVillage Dataset (public, open-source). • 5 crops (apple, corn, grape, potato, tomato) with 25 disease types. • Images resized, with augmentation. • ~37,572 training and 10,359 testing images.	• Improved MobileNetV2 (MobileNet-RepMLP). • Combines RepMLP and ECA attention (for feature weighting). • Uses Hardswish activation instead of ReLU6. • Implemented in PyTorch / TensorFlow / Keras.	• Tackles low-resource crop disease identification. • Reduces parameters and computation by ~59% vs. MobileNetV2 • Boosts accuracy using attention and feature re-parameterization	Overall accuracy : 99.53% • Precision/Recall : ≈99.5% each • 8.5% faster inference speed than MobileNetV2 • Model params: 0.91M (very compact)	• Trained on controlled images • May underperform in real-world complex scenarios • Focused only on leaf-based visual symptoms

Year	Name	Data Set	Model / Library	Findings	Accuracy	Limitations
7) 2024	**MC-ShuffleNet V2: A lightweight model for maize disease recognition**	• 2,725 maize disease images from northern Anhui, China, in a natural field • Augmented dataset to 10,845 images using methods(random rotation and flips) • Images were resized	• MC-ShuffleNetV2, ShuffleNetV2 1x • Incorporates (CBAM) Convolutional Block Attention Module • Mish activation function applied • TensorFlow-gpu 1.14 + Keras 2.2.4 is used	• Addresses problem of model bloat and high resource consumption • Focuses on accurate feature extraction. • 5×5 convolution kernel extracts more detailed features.	Overall accuracy : 99.86% • Model has 873,936 parameters and 1,751,286 FLOPs • Outperforms several models in accuracy and size	• Dataset is limited to only 6 typical maize diseases. • Model's capabilities are restricted by size and types of training samples.
8) 2025	**Shuffle-PG: Model for retrieving images of plant diseases and pests with deep metric learning**	• 126,900 images of diseases and pests. • Disease dataset with 37 classes and 87,151 images, and a pest dataset with 34 classes and 39,749 images • Classes categorized based on crop-disease/crop-pest pairs.	• Shuffle-PG, combines ShuffleNet v2 with pointwise group convolution. • Model fine-tuned using deep metric learning with contrastive loss and an online hard negative pair selection strategy. • The Adam optimizer was used	• Tackles deploying large models for plant disease diagnosis • Proposes content-based image retrieval (CBIR) approach • Deep metric learning enhances model's ability to learn discriminative features	Overall accuracy : Disease dataset -97.7% Pest dataset - 98.8% • The recall at rank 5 98.9% for diseases 99.4% for pests • 1.26 million parameters and 151.7 million FLOPs	• Dataset is imbalanced and contains a large variation in images per class. • Real-world challenges like latency, battery life, and UI design were not fully explored

Year	Name	Data Set	Model / Library	Findings	Accuracy	Limitations
9) 2025	**SqueezeNet-Based DL Framework for Accurate Tomato Leaf Disease Diagnosis & Classification**	• Public Plant Village dataset from Mendeley Data • 14,531 RGB images of tomato plant leaves, covering 9 diseases and one healthy category • The dataset has a class imbalance • Images were resized	• DL framework based on the SqueezeNet architecture • Transfer learning with a pre-trained SqueezeNet model on the ImageNet dataset • optimizers (SGDM, ADAM, RMSProp) • Implemented in MATLAB and Python	• Need for an efficient, high-accuracy, and deployable model for tomato leaf disease detection • Investigates how different training parameters impact model's performance	Overall accuracy : 99.91% • 100% recall for the healthy class (TH) during testing with ADAM F1 - score : 99.42%	• Needs to accommodate scenarios involving multiple diseases on a single leaf • Dataset may not fully represent real-world environments.
10) 2023	**BananaSqueezeNet: Lightweight CNN for the diagnosis of 3 prominent banana leaf diseases**	• Banana Leaf Spot Disease (BananaLSD) from banana fields in Bangladesh • 937 images across four classes: healthy, Pestalotiopsis, Sigatoka, and Cordana • Training set was augmented to 400 images per class	• Uses SqueezeNet's architecture, which employs "Fire modules" • Optimized using Bayesian optimization to find the best hyperparameters. • Transfer learning was applied using a model pre-trained on the ImageNet dataset	• Successfully diagnoses 3 prominent banana leaf diseases and shows generalizability of model to 7 other diseases. • Model's small size makes it suitable for real-time use with a smartphone app	Overall accuracy : 96.25% • outperformed other models (EfficientNetB0, MobileNetV3, ResNet-101) • model's size-4.78 MB • inference time-17.84 sec	• Dataset is relatively small • Study suggests future work should include instructing farmers on how to take action based on the disease severity

Year	Name	Data Set	Model / Library	Findings	Accuracy	Limitations
11) 2024	**Approach to Lightweighting Backbone Network Models through Quantization and Bayesian Optimization**	• COCO dataset used for training and validation • 5000 images used specifically for inference evaluation • Dataset supports autonomous driving scenarios	• .MobileNetV3 Small (baseline) • Modified MobileNetV3 with Leaky ReLU • Bayesian Optimization for hyperparameters • Dynamic Quantization (INT8)	• .Quantization reduced model size by ~50% • Accuracy improved by 1% • Memory usage reduced by ~97.8% • Faster inference after quantization	• Baseline: Standard MobileNetV3 • Improved: +1% accuracy after optimization and quantization	• .Leaky ReLU increases computational cost • Quantization may cause precision loss • Performance depends on careful layer selection • Tested only on COCO dataset
12) 2022	**Classification of Ear Imagery Database using Bayesian Optimization based on CNN-LSTM Architecture**	• .Public Ear Imagery Database • 880 otoscopic images • 4 classes: Normal, Myringosclerosis, Earwax plug, COM • Image resolution: 420 × 380 pixels • Collected from Universidad de Chile hospital	• .EfficientNetB0 (CNN) for feature extraction • BiLSTM for classification • CNN-LSTM hybrid architecture • Bayesian Optimization for hyperparameter tuning • MATLAB 2020 implementation	• .CNN-LSTM outperformed CNN-only model • Bayesian optimization improved hyperparameter selection • Significant reduction in training time (25 min vs 637 min) • Effective in assisting otolaryngologists	• 100% accuracy on testing dataset • CNN-only accuracy: 86.3% • Sensitivity: 100% • Specificity: 100%	• .Small-scale dataset (only 880 images) • Limited to one public ear database • Performance not tested in real clinical environments • No cross-database validation performed

EXECUTION

- **Data Collection:** Gather images of healthy and diseased crop leaves.
- **Preprocessing:** Resize, normalize, and augment data for better training performance.
- **Model Training:**
  - **YOLO (Tiny / YOLO-Nano):** Detect and localize diseased regions.
  - **EfficientNet, MobileNet, ShuffleNet, SqueezeNet:** Classify disease type and severity.
- **Integration:** Combine detection and classification into a single automated pipeline.
- **Edge Optimization:** Prune and quantize models for deployment on mobile and edge devices.
- **Testing & Validation:** Evaluate accuracy, precision, and recall to ensure reliability.
- **Deployment:** Enable real-time disease detection using drones or smartphones in the field.

SUMMARY

Lightweight deep learning models enable fast and accurate crop disease detection, bridging the gap between AI research and real-world agricultural use.

This system empowers farmers with timely insights and promotes sustainable farming practices.

Thank You