Chroma-Sense: A Memory-Efficient Plant Leaf Disease Classification Model For Edge Devices

Presenter: Kiran Kumar Kethineni

Kiran Kumar Kethineni[1], Samuel Y. Wu[2],
Saraju P. Mohanty[3], E. Kougianos[4]

University of North Texas, USA.

Email: kirankumar.kethineni@unt.edu[1], samuelwu@my.unt.edu[2], saraju.mohanty@unt.edu[3], elias.kougianos@unt.edu[4]

Outline

- The Big Picture
- Novel Contributions
- Related Works
- Proposed Solution
- Implementation and Results
- Conclusions and Future Work

Evolution of Smart Agriculture

Traditional Agriculture

- Manual labor, experience-based decision-making, simple tools.

Green Revolution

- Synthetic fertilizers, pesticides, high-yield crop varieties .

Precision Agriculture

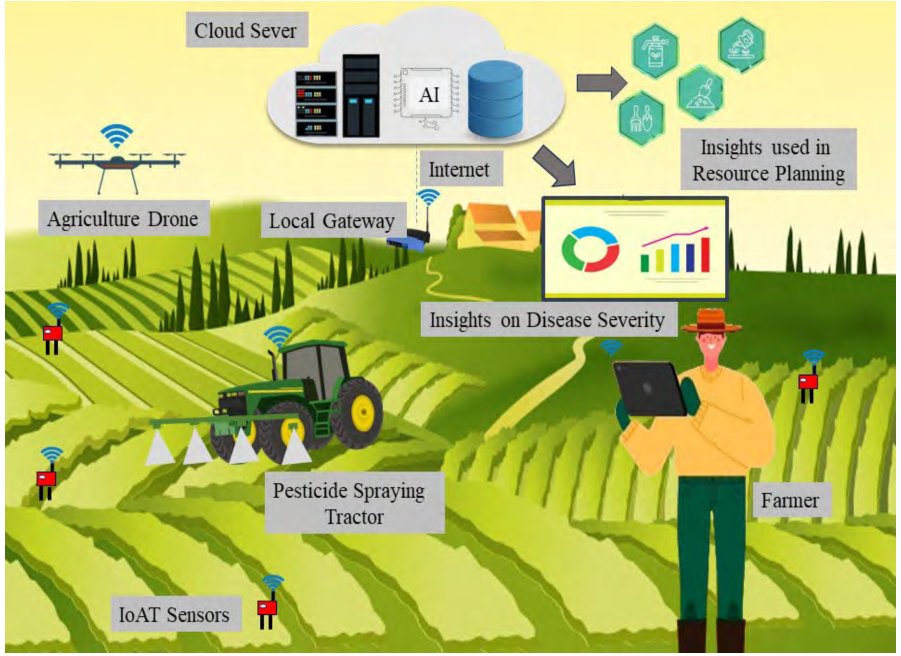
- sensors, and data analytics for automation.

Smart Agriculture

- IoT, big data analytics , and machine learning.

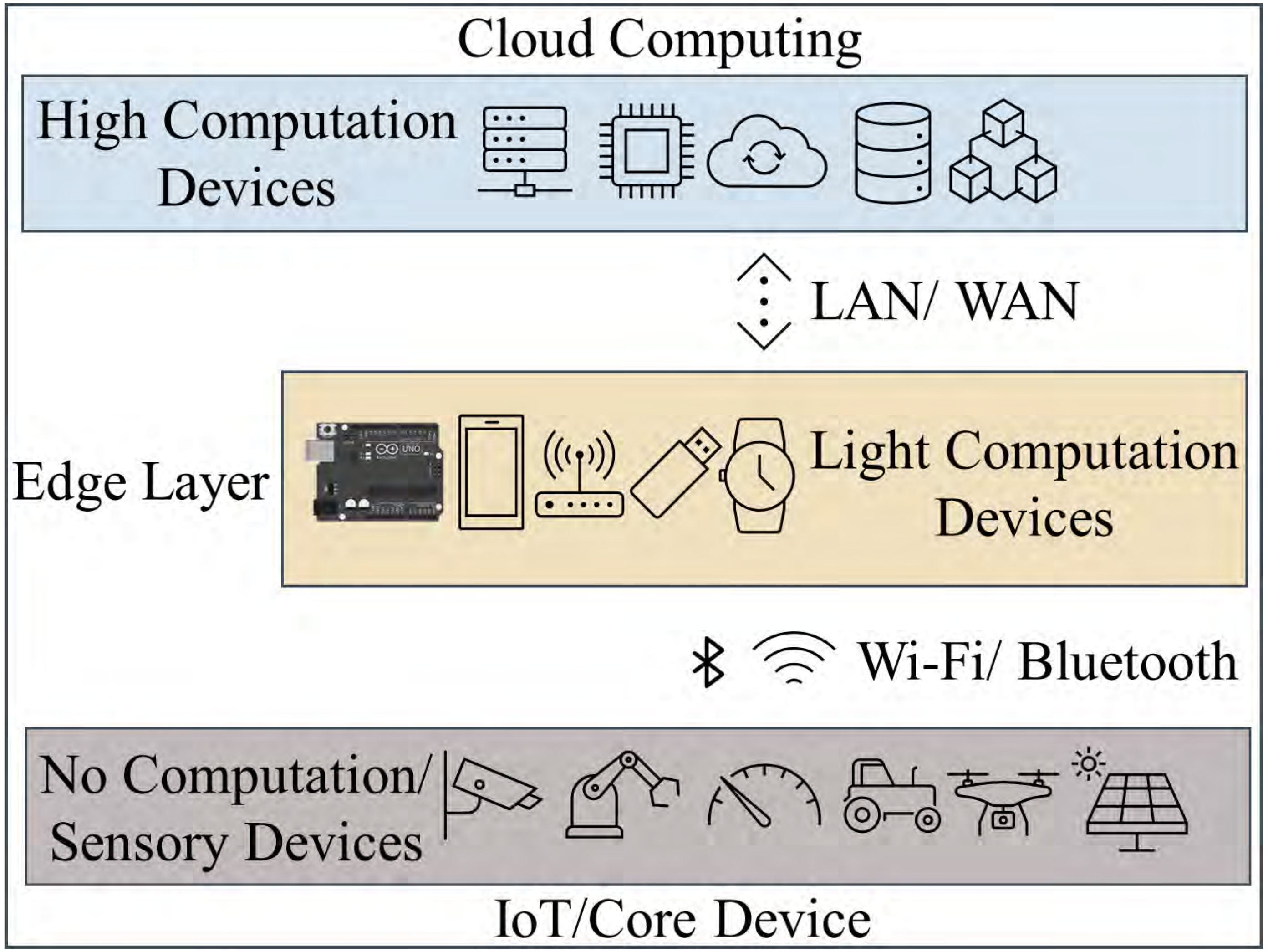
Agricultural Cyber-Physical Systems (ACPS)

- ACPS: A system that integrates physical entities, sensors, and digital technologies for real-time monitoring and control of agricultural processes.
- Plant diseases lead to economic losses, need timely intervention.
- ACPS facilitates early disease detection using computer vision.
- Continuous monitoring via ACPS allows for effective damage control.



Disease management ACPS

3 Layered structure of IoT



Edge Computing – The Challenge

Reason to adopt	Opposing constraint
Real-time data processing	Limited computational resources
Reduced latency	Energy efficiency requirements
Enhanced reliability in limited connectivity	Data management and filtering challenges
Improved data security	Heterogeneous hardware compatibility
Cost savings	Model optimization challenges

Related Works

Work	Year	Method adopted	Remark
Mohanty et al.	2016	AlexNet, GoogLeNet	Models are generic and not optimized for edge devices.
Archana et al.	2023	ResNet-50	Models are generic and not optimized for edge devices.
Peker	2021	Multi channel network ensemble	Cumulative features across R,G,B channels are not considered.
Rakib et al.	2024	A reduced CNN model with layer quantization	Layers are reduced and optimized for edge devices, but not the architecture.

Related Works

Work	Year	Method adopted	Remark
Bedi and Gole	2021	Convolutional auto-encoder	Optimized for plant diseases but limited to binary classification.
Chowdhury et al.	2021	EfficientNet	Models are optimized for edge devices but not for plant diseases.
Ashwinkumar et al.	2022	MobileNet	Models are optimized for edge devices but not for plant diseases.
Chroma-Sense	2025	MobileNet style parameter reduction with serial multi channel processing	Models are optimized for edge devices and for plant diseases.

Proposed Solution

- Plant diseases typically manifest as spots, patches, textures on leaves, often characterized by distinct colors.
- Their distinguishing features are relatively simple, with different colors
  - Patches
  - Mosaics
  - Powdery textures
  - Velvety surfaces
  - Leaf curls
  - Circles
  - Stripes

Apple Scab

Apple Cedar Rust

Proposed Solution

- Since the features are simple and same across different colors, they can be simple CNN network, processing R,G,B independently.

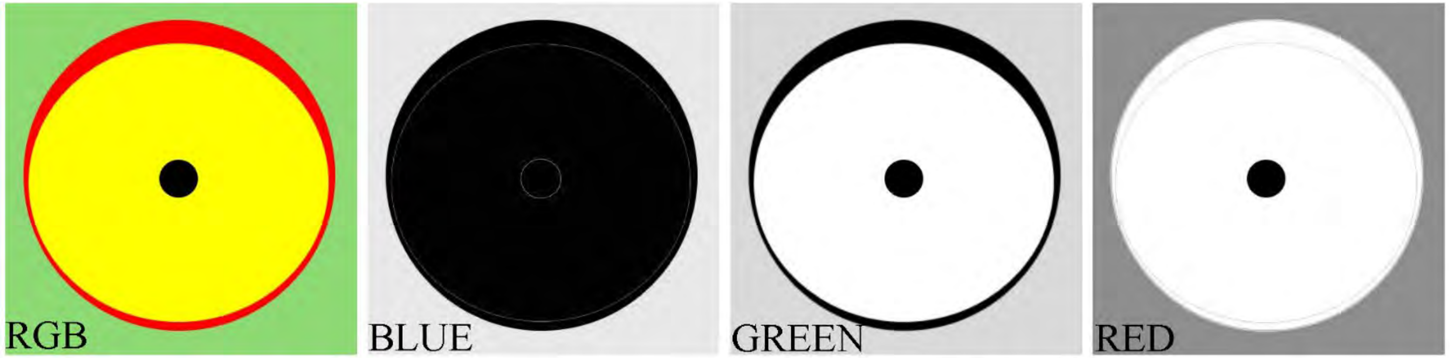


Illustration of features of Apple Ceder Rust disease.

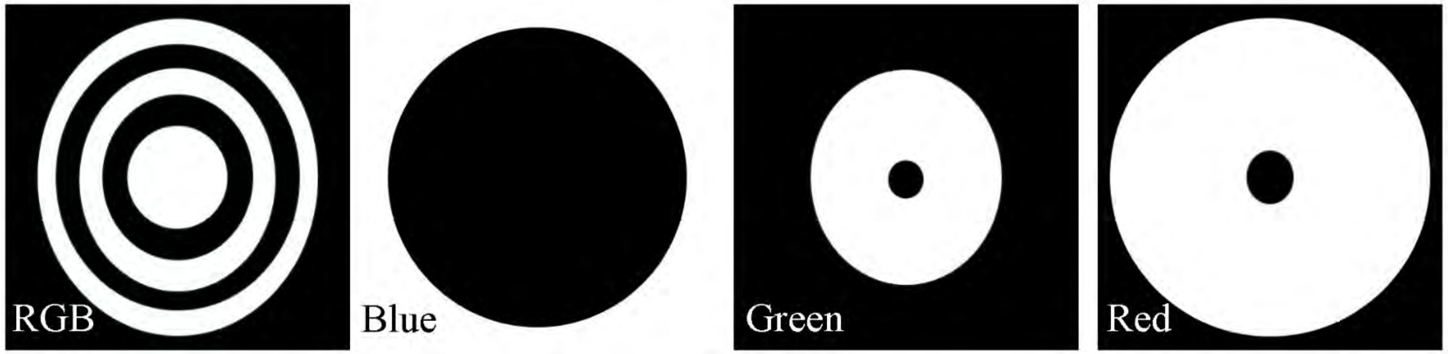
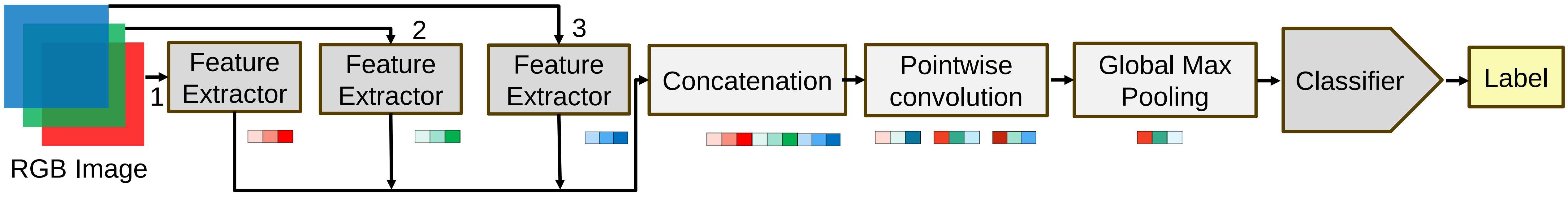


Illustration of features needed to be learned.

Novel Contributions

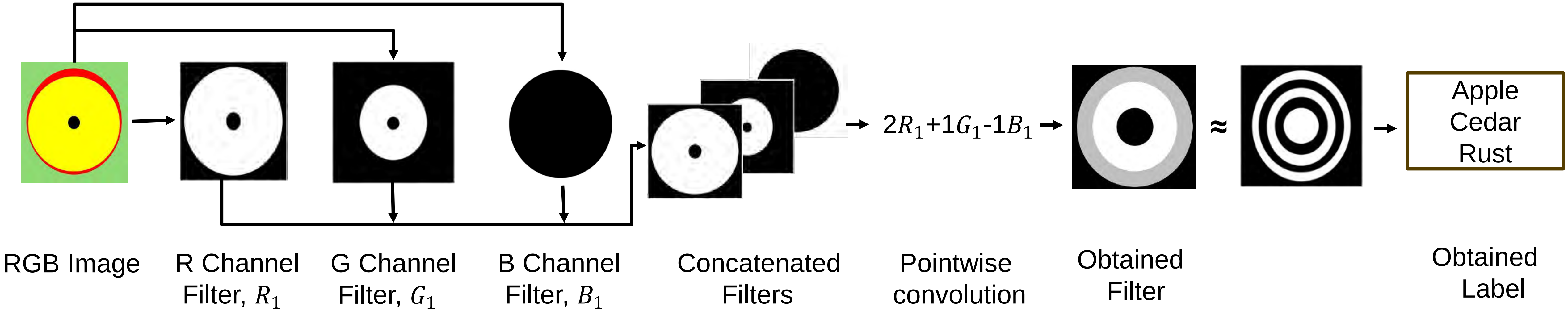
- Memory Efficiency: Processing the R, G, and B channels in the image individually and sequentially reduces the width of the CNN model, significantly decreases the RAM required.
- Reduction in parameter count: The reuse of the CNN feature extractor across all color channels, reduces the parameter count.
- Simplified Semantic Feature Extraction: As the CNN network processes grayscale images of each channel, it is compelled to learn shapes and textures.
- Explainability: By extracting channel-specific features and stacking them for classification, spatial information is embedded, making the decision-making process transparent and explainable.

Proposed Method

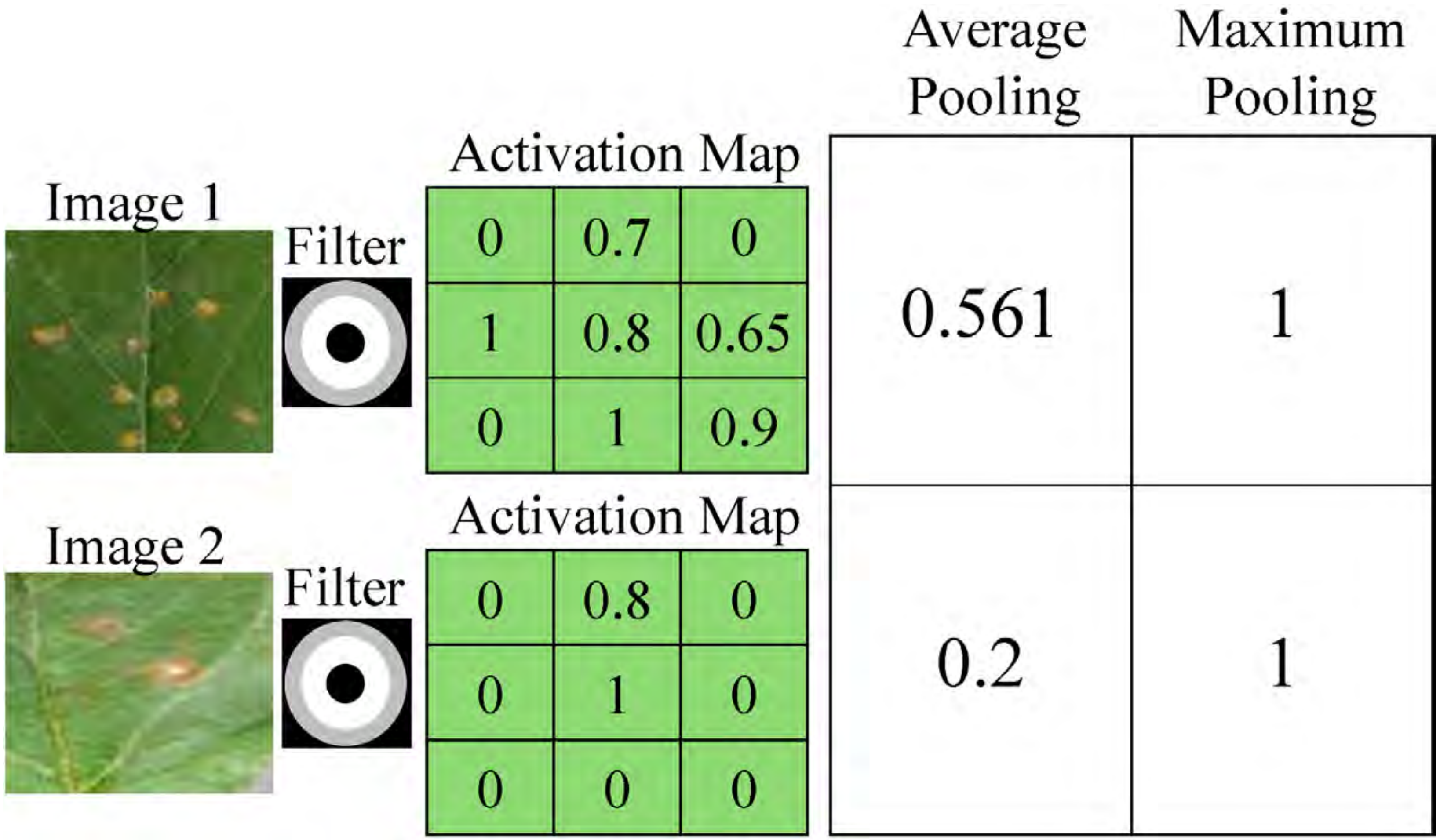


Proposed Architecture

Illustrative Working

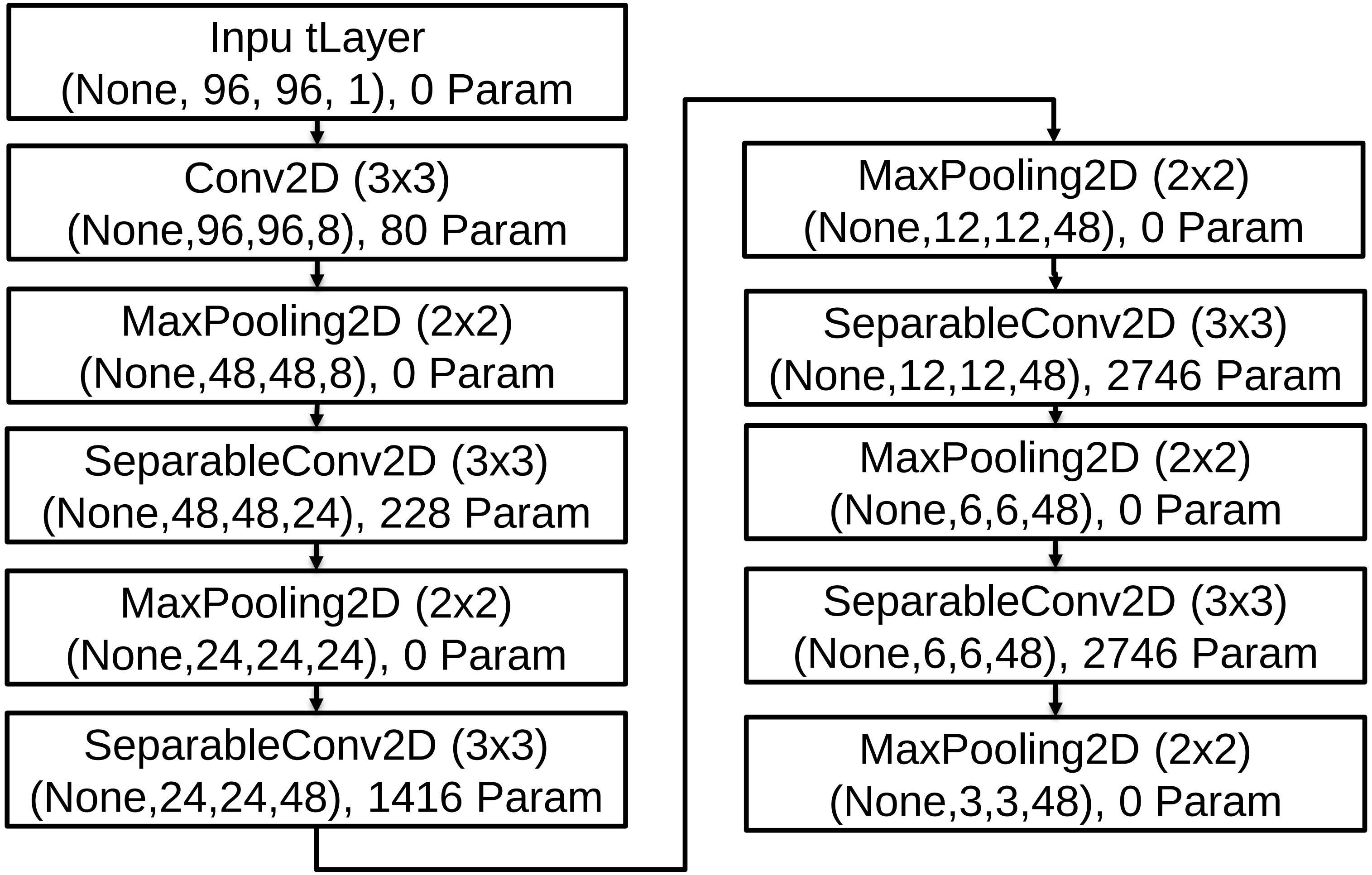


Proposed Method



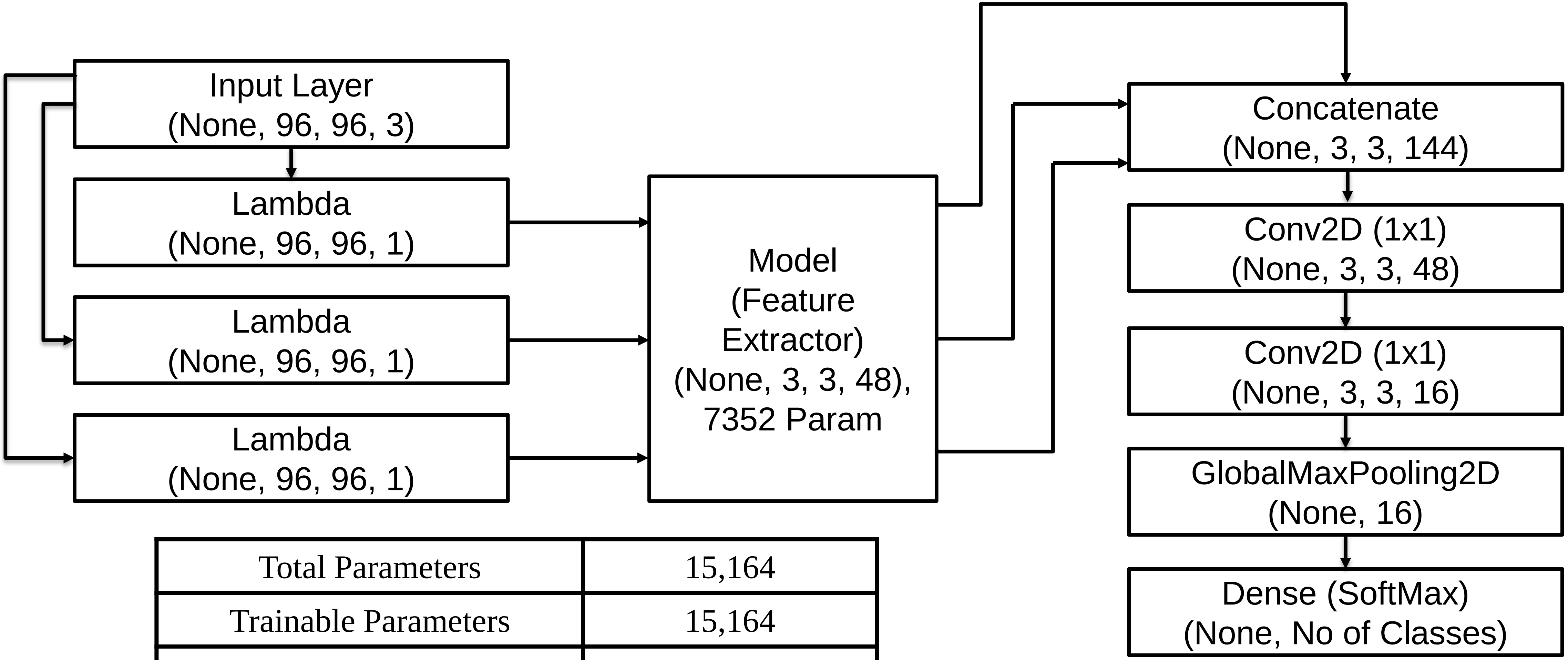
Comparison between average and max pooling.

Proposed Feature Extractor



Total Parameters	7,352
Trainable Parameters	7,352
Non-trainable Parameters	0

Proposed Classification Model

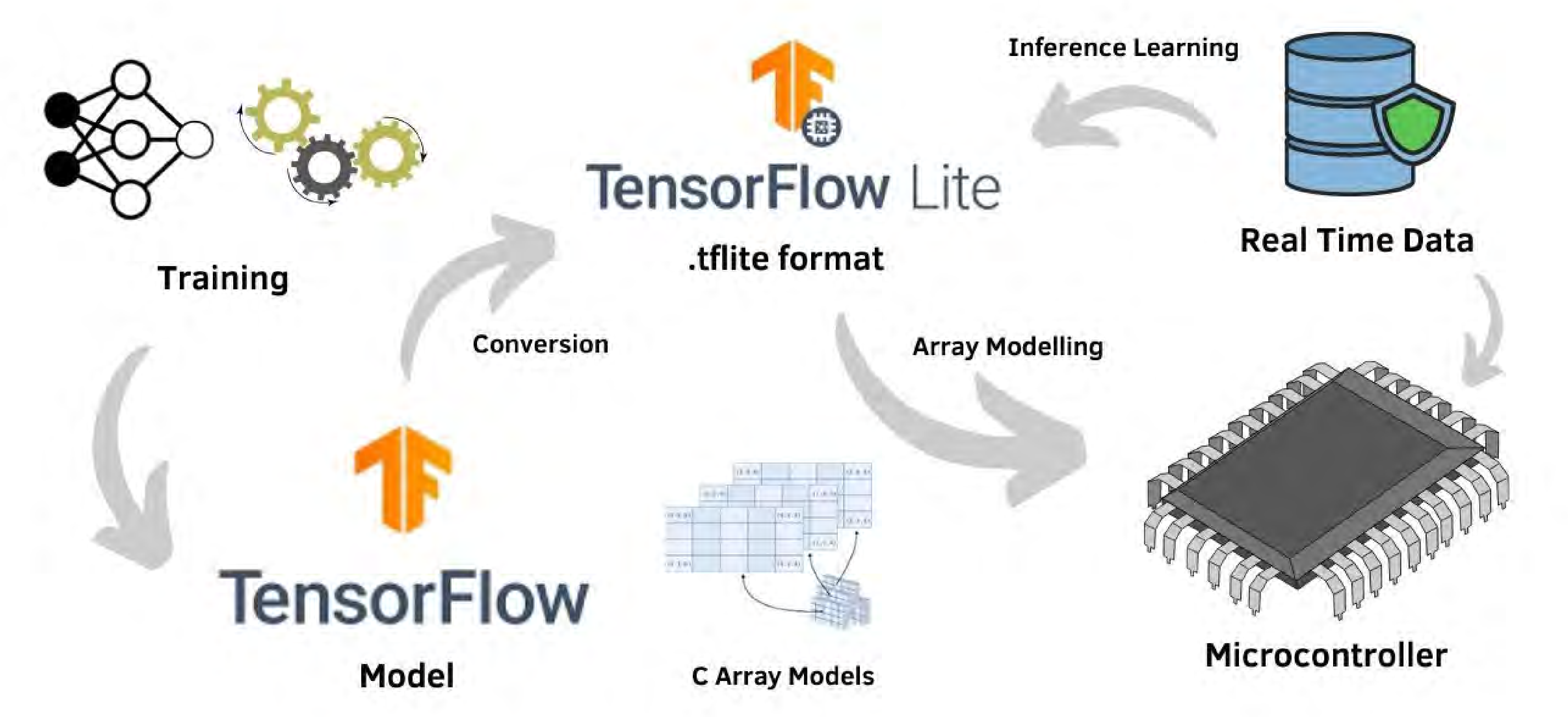


Total Parameters	15,164
Trainable Parameters	15,164
Non-trainable Parameters	0

Implementation and Results

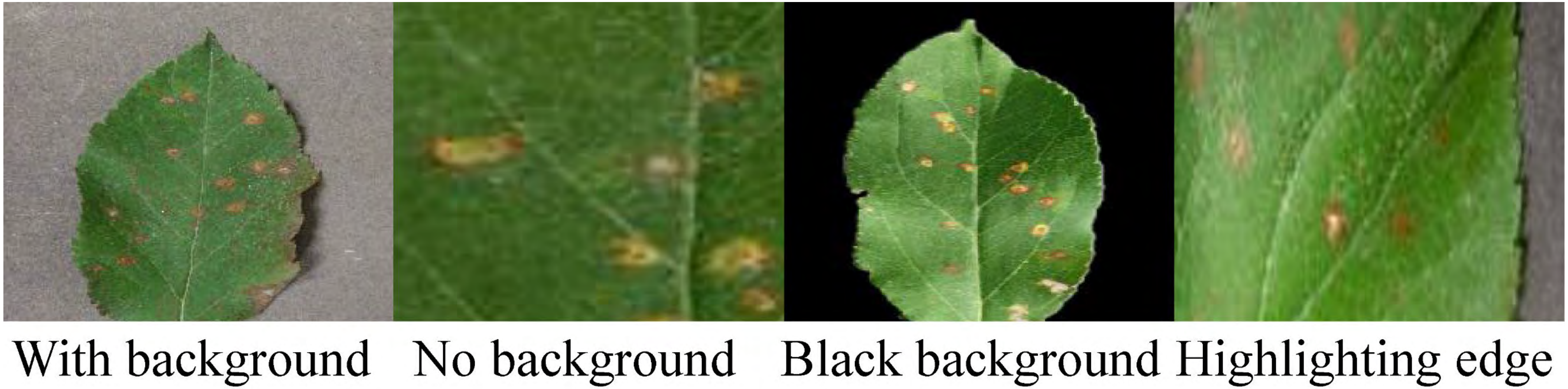
- The proposed Chroma-Sense was developed using Python and TensorFlow for the CNN classification model.
- TensorFlow Lite used to quantize the model to Int8.
- It was individually validated on 4 plant types, using a total of 18,000 images across 18 classes from the PlantVillage dataset.
- The results of the developed models, tested on 3 different edge devices : OpenMV H7, OpenMV H7 Plus, Arduino Nicla Vision.

Implementation and Results



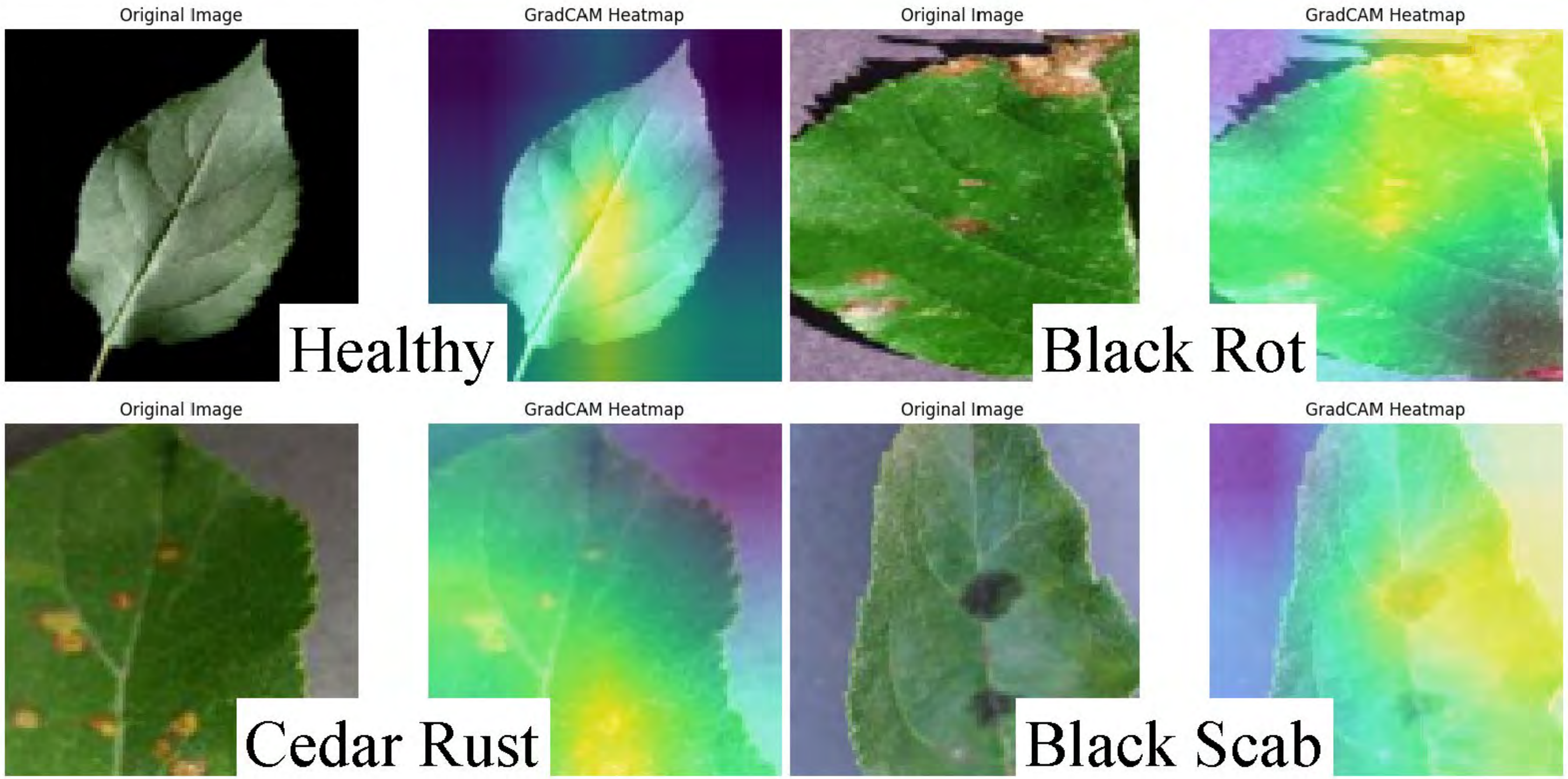
The TensorFlow Lite Micro workflow. Image used courtesy of Saumitra Jagdale

Implementation and Results



Dataset created for training

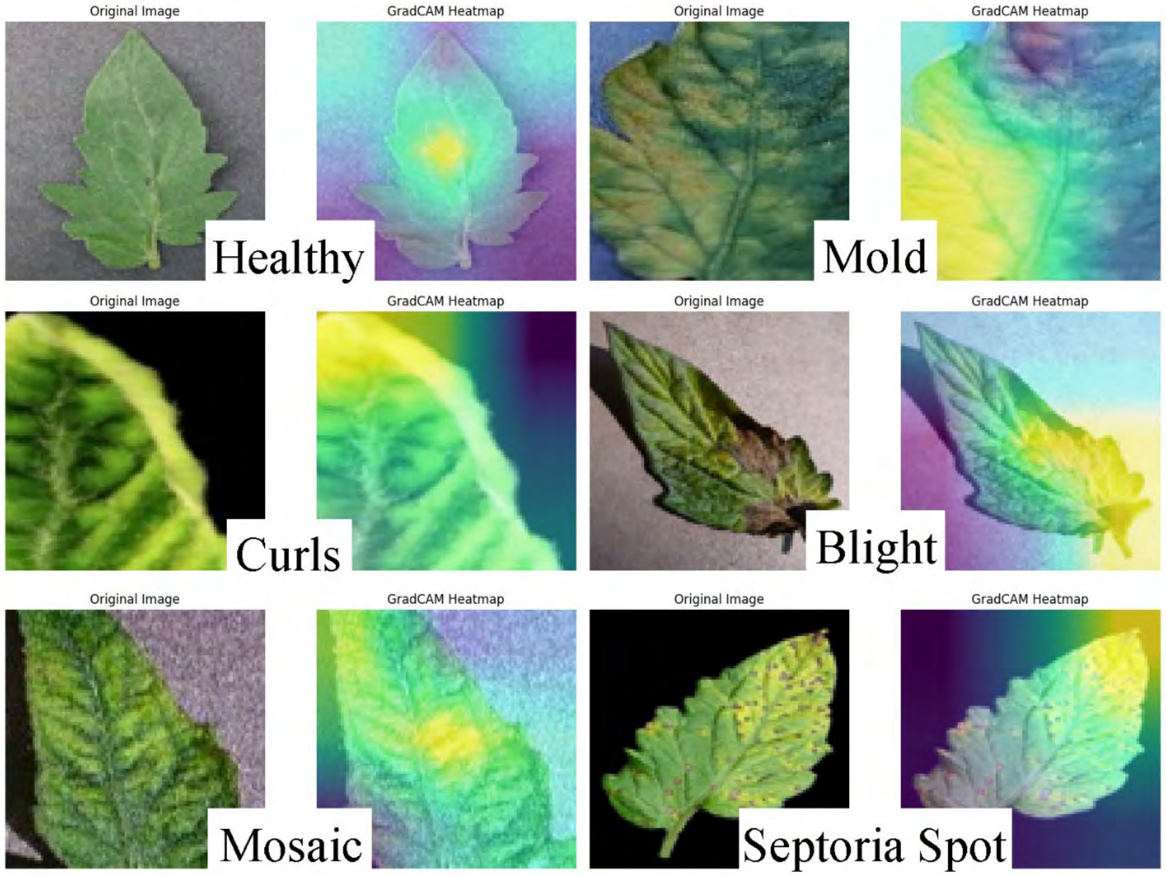
Implementation and Results



Class	F1 Score
Healthy	0.93
Black Rot	0.93
Black Scab	0.95
Cedar Rust	0.91

Grad-CAM results for Apple Dataset (Unoptimized Model)
93% Accuracy (TFLite, Int8)

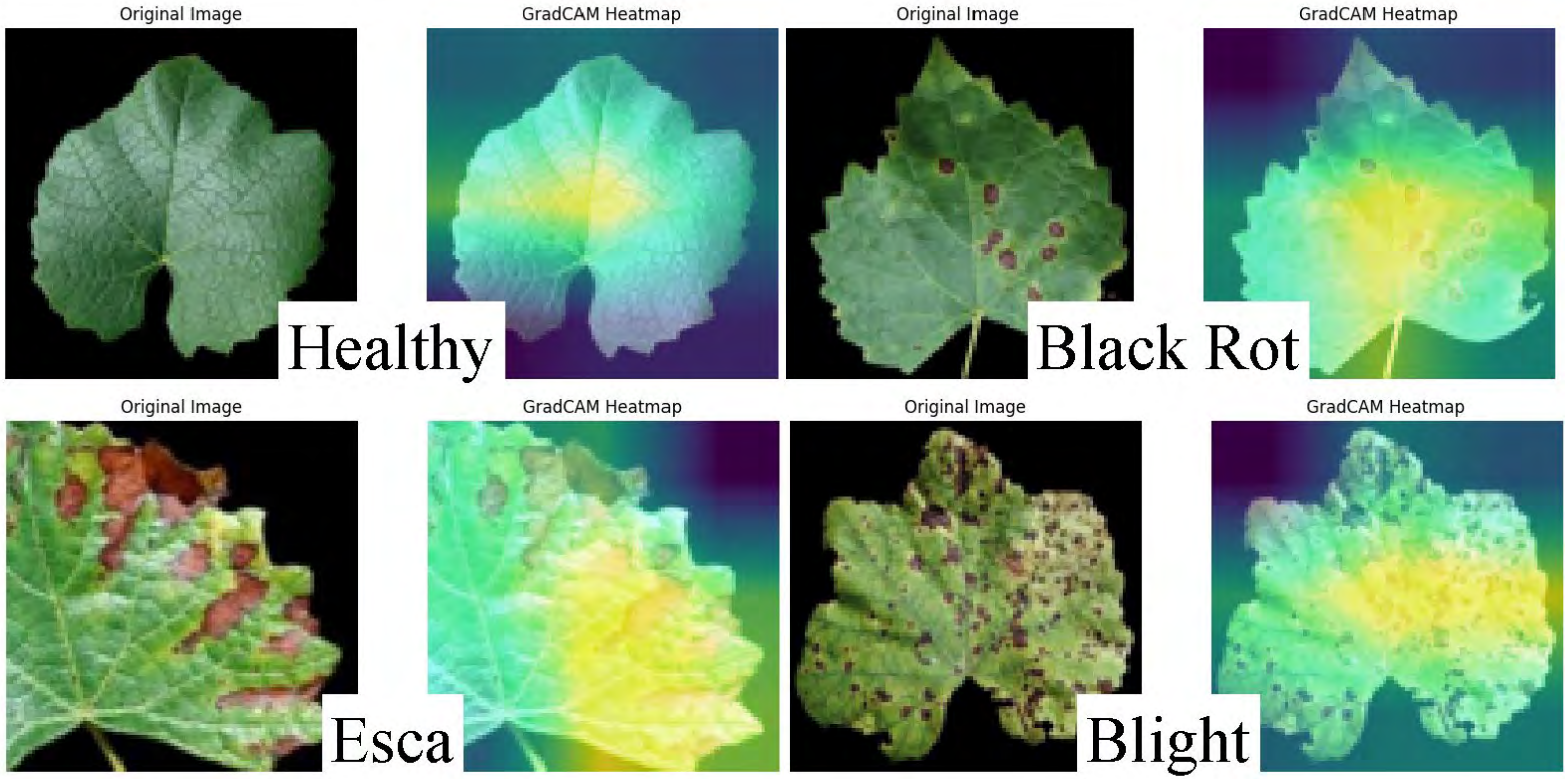
Implementation and Results



Grad-CAM results for Tomato Dataset (Unoptimized Model)
89% Accuracy (TFLite, Int8)

Class	F1 Score
Healthy	0.90
Mold	0.88
Curls	0.88
Blight	0.91
Mosaic	0.90
Septoria Spot	0.87

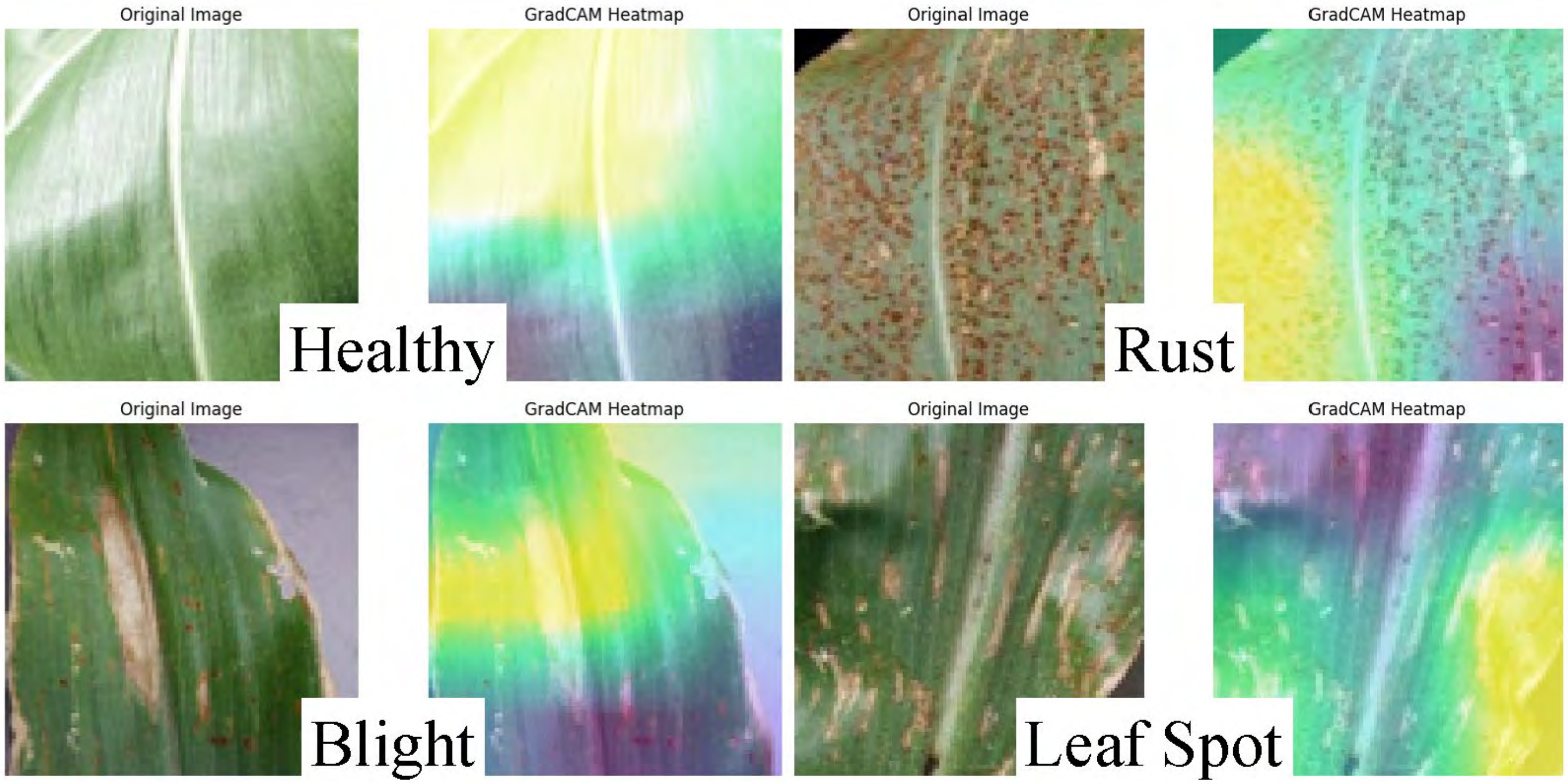
Implementation and Results



Class	F1 Score
Healthy	0.97
Black Rot	0.95
Esca	0.96
Blight	0.96

Grad-CAM results for Grape Dataset (Unoptimized Model)
96% Accuracy (TFLite, Int8)

Implementation and Results



Class	F1 Score
Healthy	0.94
Rust	0.91
Blight	0.92
Leaf Spots	0.95

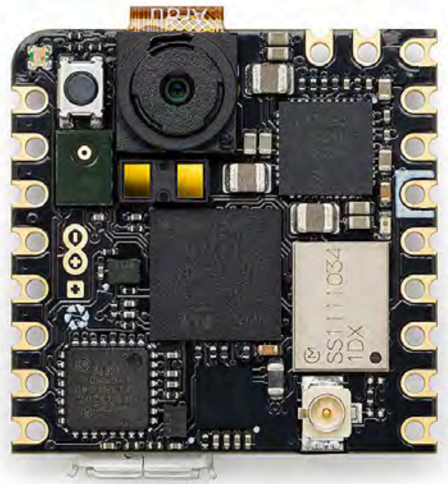
Grad-CAM results for Corn Dataset (Unoptimized Model)
93% Accuracy (TFLite, Int8)

Implementation and Results

Edge Device	Heap Memory	Flash Memory	Processor
OpenMV H7	256KB	128KB	Cortex-M7
OpenMVH7 Plus	4MB	32MB	Cortex-M7
Arduino Nicla Vision	256KB	16MB	Cortex-M7 + Cortex-M4

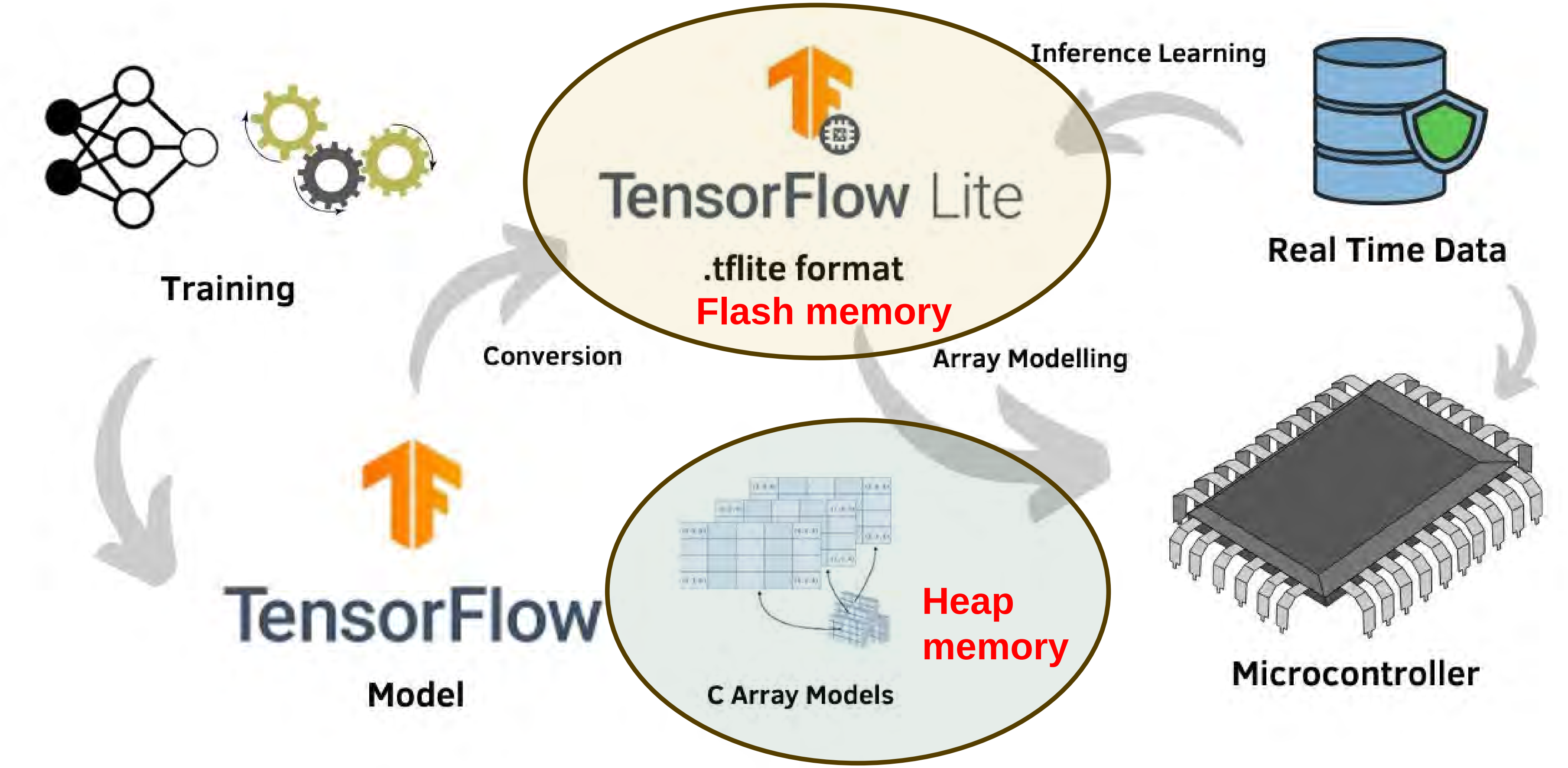
OpenMV H7

OpenMV H7 Plus



Arduino Nicla Vision

Implementation and Results



The TensorFlow Lite Micro workflow. Image used courtesy of Saumitra Jagdale

Implementation and Results

Model	Acc.Int8		RAM	Flash Memory	Parameter Count	Madds	Frames/Sec		
	Apple	Tomato					H7	H7 Plus	Nicla Vision
Conv2D	95.2	88.8	336KB	502KB	491K	130M	NA	1.7	NA
Grouped Conv	86.4	71.7	336KB	187KB	169K	43M	NA	4.5	NA
MobileNet	86.7	86.6	341KB	99KB	66K	21M	NA	5.3	NA
MobileNetV2	95.4	90.4	271KB	203KB	153K	25M	NA	7	NA
MobileNetV3	92.8	92.6	216KB	138KB	97K	20M	NA	5.3	7.6
EfficientNetV2	91.8	88.6	240KB	160KB	110K	27M	NA	5.1	6.9
SqueezeNet	97.2	83.5	338KB	80KB	56K	23M	NA	6.4	NA
ShuffleNet	90.2	83	347KB	129KB	88K	13M	NA	2.3	NA
Sque. and Exci.	82.2	79.1	149KB	550KB	527K	32M	NA	5.6	NA
Multi.Ch. CNN	88.8	86.1	160KB	87KB	62K	43M	4.7	4.7	4
Chroma-Sense	93.1	89.3	160KB	54KB	15K	8.3M	8.4	8.4	7.2

Implementation and Results

Model	Enhancement
Conv2D	Standard Conv2D layers with no optimization.
Grouped Conv	Reduced parameters through group convolution.
MobileNet	Reduced parameters, RAM by using depthwise separable convolution.
MobileNetV2	Improved expressive power while reducing feature map size and memory usage.
MobileNetV3	Attention mechanisms helped reduce feature maps and memory usage, parameters.
EfficientNetV2	Fused-MBConv improved training speed and increased computations.
SqueezeNet	Fire modules reduced parameters but required more memory for expansion.
ShuffleNet	Feature cross-talk improved expressive power and reduced parameters, but did not save memory.
Sque. and Exci.	Attention mechanisms enhanced expressive power and reduced memory usage, but higher computations.
Multi.Ch. CNN	Serial processing helped reduce memory usage, parameters, and computational requirements.
Chroma-Sense	Reduced memory usage, parameters, and computations with serial processing and separable convolutions.

Conclusion and Future Work

- Chroma-Sense is designed for 96x96 images and performs classification based on features within the receptive field.
- This low resolution may fail to capture fine details in images with multiple leaves or crops.
- It fails to capture global leaf-level features, leading to misclassification when multiple plant types are present in the dataset.
- Future work should focus on enhancing the feature extractor’s representational power and designing efficient memory allocation algorithms to handle higher resolutions.

Thank You !!