Lite-Agro: Exploring Light-Duty Computing Platforms for IoAT-Edge AI in Plant Disease Identification

Catherine Dockendorf[1][0009−0008−7778−9210], Alakananda Mitra[2][0000−0002−8796−4819], Saraju P. Mohanty[1][0000−0003−2959−6541], and Elias Kougianos[3][0000−0002−1616−7628]

[1] Department of Computer Science and Engineering, University of North Texas, USA.
catherine.dockendorf@unt.edu, saraju.mohanty@unt.edu
[2] Nebraska Water Center, Institute of Agriculture and Natural Resource, University of Nebraska-Lincoln, USA. amitra6@unl.edu
[3] Department of Electrical Engineering, University of North Texas, USA.
elias.kougianos@unt.edu

Abstract. The Lite-Agro study aims to deploy deep learning neural network models for pear disease identification through tree leaf image analysis on TinyML device. A case study on pear leaves is conducted with publicly available pear disease dataset. Quantitative comparisons are made between different datasets. *Lite-Agro* is a light-duty image computing detection solution that is tested for deployment on a microcontroller. The novelty of *Lite-Agro*, lies in the export of a lightweight TinyML, Tensorflow Lite model that is geared for low power applications on battery powered hardware. The goal is to find the best model that is custom selected for the application and achieves the highest accuracy. The study emphasizes finding a balance between size, accuracy and performance. In future iterations of the study, Lite-Agro is to be mounted on an unmanned aerial vehicle to be powered with solar panels. Modern low powered microcontroller devices are to be a staple implementation in Smart Villages.

Keywords: Smart Agriculture · Agriculture Cyber-Physical System (A-CPS) · Internet of Agro Things (IoAT) · Smart Village · TinyML · Edge-AI · Plant Health · Plant Disease

1 Introduction

Deep Learning has made major advancement in the last ten years. They are being employed in applications encompassing a wide range of use cases from speech generation, text processing and image identification. In 2020, the world wide yield of pear was 23 metric ton. However, pear diseases can adversely affect the pear yield. Early and automatic detection of pear diseases can stop over use of herbicides, reduce the cost related to the expensive expert services to detect the diseases, and mitigate the disease early to reduce the financial loss of the farmers.

When deep neural networks (DNN) are employed to detect plant diseases, they automatically extract the features from the input data and detect the disease. DNN provides high accuracy after training with a large dataset. Images of the pear leaves can be used as the input data. However, this current research, *Lite-Agro*, aims to bring the solution to the farmer with a low power edge solution. This research on TinyML [20] and it's application on micro controllers branches off from a plant disease identification [14] and crop damage estimation [15] study. *Lite-Agro*, applies deep learning to improve efficiency in Smart Agriculture [16]. *Lite-Agro* is designed to be a lightweight system that runs on a low power microcontroller and aims to find optimization in terms of model accuracy while being a low cost hardware implementation.

The paper is currently organized into various sections. Section 2 summarizes novel contribution. Section 3 discusses prior work. A system level overview is discussed in Section 4. Section 5 talks about proposed training methods and a comparative perspective. Experimental validation is discussed in Section 6. Section 7 concludes the article with an overview of future work.

2 Novel Contributions of the Current Paper

2.1 Problem Addressed and Proposed Solution

Lite-Agro, a lightweight, low power TinyML-based pear disease identification system, is proposed to address the needs of a plant disease solution in pear farms. Manual inspection of leaves for plant diseases translate to time inefficiencies. A combination of various deep learning models were trained until the best recognition accuracy was attained. The novelty lies in the application of compact modern Convolutional Neural Networks (CNNs) that are evaluated on a publicly available dataset [6]. The CNN-based model Xception generated the highest recognition accuracy of 99.97% however due to microcontroller memory constraints, other models are being considered. This is important as a computing gear for smart villages.

2.2 Novelty and Significance of Proposed Solution

The novelty of Lite-Agro lies in the exploration of low powered and lightweight hardware platform for the optimum TensorFlow [29] Lite model, trained to learn the identification of diseases in pear leaves. The study looks at how modern deep learning convolutional neural networks contribute to the optimization and reliability of TinyML [8] applications. The exploration of a light-duty computing hardware platforms powered by TinyML: TensorFlow Lite for microcontrollers has key potential in Smart Village edge devices. Exploring TinyML options and how it can be utilized as the hardware solution for a TinyML image capture implementation, comprises as the element of novelty in this study.

3 Related Prior Works

The work in [28] has explored three classifications of infections; *Septoria piricola* [1, 25], *Alternaria alternate*, and *Gymnasporangium haracannum* [11]. DiAMOS plant study, mentions the use of pretrained models or an ensemble of pretrained models such as EfficientNetB0 [24], InceptionV3 [23], MobileNetV2 [19], and VGG19 [22]. To conclude with a model that results in the highest accuracy, the study employed training on the PDD2018 dataset with VGG16, InceptionV3, and Resnet50 [12] models. Data augmentation, number of epochs and image resolution size, were the variable parameters in the experiment. Table 1 presents a summary of these works.

Alternaria alternata in plant disease pathology, is defined as a type of an opportunistic fungus that is the cause of spot or discoloration on leaves. *Gymnosporangium haracannum* [11], more commonly called as juniper rusts, is an orange lesion and rust like spotting on a leaf. The disease is more of a cosmetic eyesore since infected fruits can look like corona gelatinous fingers. *Septoria pyricola* [1], is another plant disease found on Pears (Pyrus Communis). Outbreaks have been said to have occurred in pear orchards. A simple description of this disease is a brown outer rim leaf spot with a white central lesion.

The study conducted in [5] benchmarks a comparison between an ensemble combination of three neural networks such as EfficientNetB0 [24], MobileNetV2 [19], and InceptionV3 [23]. EfficientNetB0 + InceptionV3 [12], produced the highest value of recognition accuracy at 91.14%.

Table 1. Accuracy of CNN and Ensemble-based Models in the PDD2018 *(left)* DiaMOS Study *(right)*.

PDD2018 Study	Accuracy	DiaMOS Study	Pixel Size	Accuracy
EfficientNetB0 citeefficientnetB0	89.02 %	VGG16	224 x 224	78.34 %
InceptionV3 [23]	84.44 %	VGG16	600 x 600	96.80 %
MobileNetV2 [19]	87.70 %	InceptionV3	224 x 224	80.23 %
EfficientNetB0 + InceptionV3	91.14 %	InceptionV3	600 x 600	97.99 %
EfficientNetB0 + MobileNetV2	86.21 %	ResNet50	224 x 224	73.85 %
InceptionV3 + MobileNetV2	85.35 %	ResNet50	600 x 600	98.70 %

CNN-based plant disease identification is presented in [13]. In [26], a study that incorporates the novel use of solar power in the hardware aspect of plant disease identification has been presented.

4 LITE-AGRO: A SYSTEM LEVEL OVERVIEW

The system overview of Lite-Agro is shown in the Figure 1. A public image dataset is used for the training of the model. In terms of software, the research utilizes Tensorflow [7], a framework written in Python that contains C++ machine learning [21] and artificial intelligence libraries. On top of that is Keras, which is defined as a software API (Application Programming Interface), fully integrated with the backend TensorFlow. GPUs have revolutionized machine learning studies and have enabled researchers to explore and apply real time parallel processing algorithms. Using AMD's open source framework ROCm, the authors perform high performance computing experiments and maximize multi-core capabilities [17]. The model generated through Tiny ML techniques, is exported as a Tensorflow Lite model that is then converted as a C source code byte array. Tensorflow Lite is the ultra low power port of TensorFlow designed to run on microcontrollers [27]. The EspressIf firmware compiles the model and the program together. The lightweight TinyML model runs on a single-board computer, which interfaces with an edge-server board equipped with an camera.

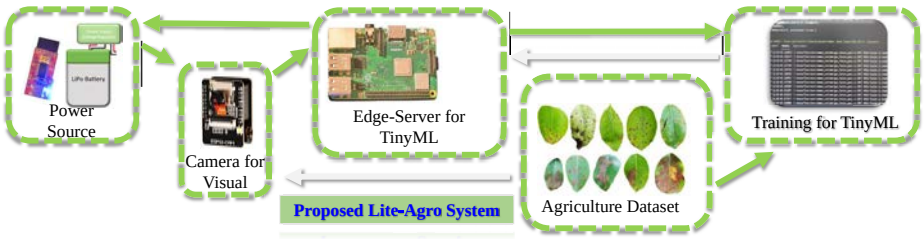

Fig. 1. System Overview of Lite-Agro.

5 PROPOSED TRAINING METHOD

5.1 Proposed Methodology

The debate whether to train using deep learning and which platform to choose, is a critical question in the *Lite-Agro* study. Deep learning is preferred, and this is attributed to the accuracy of results. The prospect of the automated predictive capability offered by deep learning is a state

of the art technology by itself, and the possibilities on the application of automation are endless. A CNN Network, the master algorithm in computer vision [18] is an imagery architecture used to process the pixels of images. Model Training is the step where a network learns from the dataset to determine's the model's weight and biases. A summary of the Tensorflow model training algorithm procedure used in this study is shown in Algorithm 1.

Algorithm 1 TensorFlowLite Model Training Procedure with Keras API.

```
Declare folder path and run Image Data Generator on training, validation and test labels.
for iteration = 1,...,3006 images do
    Preprocess images via VGG16.
    Call flow from directory and pass folder path, image size, classes and batch size.
    Preprocess image dataset from directory.
    Set parameters categorical cross-entropy and 256 x 256 image size.
    Set color mode RGB, batch size, and validation split of 0.2.
end for
for iteration = 1,...,3006 images do
    Preprocess image validation set from directory with categorical 256 x 256 image size.
    Set color mode RGB, batch size and validation split of 0.2.
end for
Declare a Sequential Model or call a Pretrained model.
Compile with rmsprop, categorical cross entropy, and set accuracy metrics.
for iteration = 1,...,100 epochs do
    Train model by calling the fit method.
end for
Save Model and Weights. Load Model.
Convert Keras Model to TensorFlowLite Model. Open file to save.
```

Epoch was set to 100 and it took 6 hours and 45 minutes to reach accuracy of 99.73%. The best performing model was produced by an InceptionV3 [23] inspired architecture,"Xception" [2] model. The Inception [23] model, is a stack of layers meant to extract features and are conceptually similar to convolutions, which are highly intensive computational processes. Keras libraries abstract the convolution process in it's library function calls. Various combinations of parameters were tested to generate the best accuracy numbers. The Xception architecture's performance is attributed to the more efficient use of model parameters. The Table 2 shows a comparative analysis between PDD2018, DiaMOS and the current paper (*Lite-Agro*). The approach was adopted in [5] to address the accuracy problem on pear disease recognition, was to adopt an ensemble CNN method during training.

Table 2. Quantitative Analysis of Current Paper with existing Pear Disease Imaging Works.

Works	Dataset	Resolution Size	Model	Recognition Accuracy
Yang, et al.[7]	PDD2018	600 x 600	Resnet50 [9]	98.7%
Fenu, at al. [3]	DiaMOS	224 x 224	EfficientNetB0 [24]+InceptionV3 [23]	91.14%
Current Paper (Lite-Agro)	DiaMOS	256 x 256	Xception [2]	99.73%

Compilation is the step to prepare the model for training. An optimizer is a learning algorithm that models the input and adjusts the network as it undergoes training. In previous

paper studies [5], most mention Adam optimizer as the default choice. Adam optimizer is a learning algorithm characterized as an extension of a stochastic gradient. It is best suited for large datasets and calculates at random points in each iteration to achieve a faster convergence. In this experimentation, the compile step argument selected was an RMSprop optimizer. In an RMSprop optimizer, gradient descent algorithms, achieving a faster learning rate as steps increments are larger, leading to a faster convergence.

5.2 Evaluation Metrics

We see how recognition accuracy increases in every epoch stage throughout the training. Tensorboard handles the logging, reporting and graph generation. Data visualization becomes easier given the set of TensorFlow tools. The graph on Fig. 2 shows the trend of accuracy increase per epoch during the training procedure.

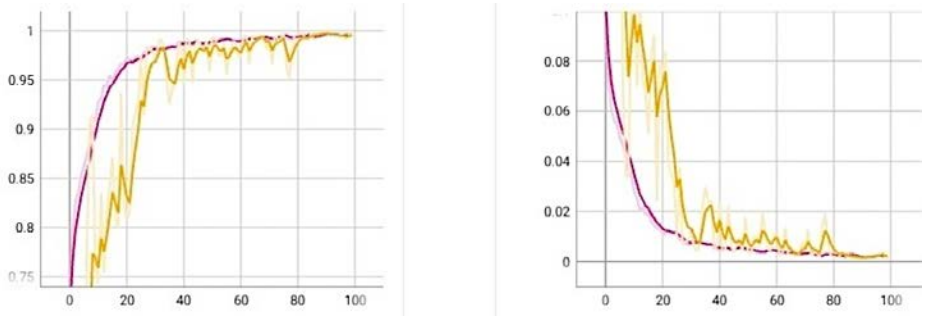

Fig. 2. Recognition Accuracy*(left)* vs Loss *(right)* at Various Epochs.

6 Experimental Results

We used DiaMOS dataset consisting of 3505 images of pear leaves of which fall under four classifications; healthy, spot, slug and curl [5]. A selected example is presented in Figure 3.

TinyML [20], short for Tiny Machine Learning, describes hardware, algorithms, and software [27]. TinyML targets being able to run inference programs on battery operated hardware and memory constrained implementation. The ability to port ML models on microcontrollers brings about countless application possibilities. Much of the emphasis in this study will be on a TensorFlow version of TinyML called, TensorFlow Lite for Microcontrollers. Keras API allows the conversion of an .h5 format of a model into a compact TFLite extension. This TensorFlow Lite conversion allows bigger models with a relatively large size, to execute in a more compact form. An Xception trained model of size $163,373$KB for example, when ran through the aforementioned TFLite converter, gets compacted down to $81,257$KB. This port of TensorFlow is optimized for running on edge devices and geared for space efficiency and addresses memory constraints. The TFLite model is once again converted to a *C* source file byte array, that describes the model. Inference is then run on the trained model and translated into the most probable set

(a) Spot and Slug Leaves from the DiaMOS Plant Dataset

(b) Curled Leaves from the DiaMOS Plant Dataset

Fig. 3. Representative of the DiaMOS Plant Dataset.

Table 3. Comparison of Tensorflow Supported Platforms.

Board	Microprocessor	CPU Clock	Voltage	SRAM Size	Connectivity
Arduino Nano 33 BLE Sense	nRF52840	64 MHz	3.3V	256 KB	USB UART, SPI, I2C, BLE, SPI
STM32F746 Discovery Kit	32bit ARM Cortex	48MHz	3V-5V	192KB	USB LQFP100 I/O
Espress ESP-EYE	32-bit ESP32	240 MHz	3.3V	8MB PSRAM	UARD, USB, BLE, SPI, I2C, WiFi
Sony Spresense	-M4F6 Core	156 MHz	3.3V-5V	1.5MB	GNSS, UART, I2C, SPI, I2S

of classification. Currently, there are a limited number of development boards supported by TFLM, mentioned on the TensorFlow website which are summarized in the Table 3.

Onboard is the Xtensa single/dual core 32 bit LX6 microprocessor [30] with 448 KB ROM, 520KB SRAM and 16KB SRAM. To program the ESP32-CAM WiFi, TensorFlow Lite uses firmware EspressIDF [4] to build and configure the board. Lastly, once the source code has been modified, the trained model is added, then the program can once again be flashed from the Raspberry Pi 3B board to the SD card and executed using a monitor call. Initially, the ESP32-CAM GPIO pins are interfaced with FTDI adapter. With the ESP32-CAM wiring set-up, the main complication lies in the need to connect the GPIO0 to ground every time the program needs to be reflashed. The ESP32-CAM takes photos at specific intervals and displays information whether the pear leaf is healthy or not through a monitor. Once it detects a diseased pear leaf, the information shall be displayed to an edge server and peripherals may be connected to display the information.

Fig. 4. IDF Monitor Pear Disease Detection Test *(left)*. Test Set-up (*right*)

7 Conclusion and Future Work

Deployment of deep learning models trained in pear disease identification, implemented on micro controllers were explored. A lightweight port of Tensorflow called TinyML [27] or TensorFlow Lite was used. The ESP32-CAM delivers a balance of price, power and performance and make a good hardware selection for edge devices. Often times, memory is severely resource constrained [3] but being able to run tiny deep learning models have great contribution in the automation of devices. This is why light-duty computing platform for IoAT-Edge devices and how it can improve the processing in Smart Agriculture processes was explored. The deployment of Lite-Agro in a proper testing environment where the pears are located and being able to gather actual field data can further contribute to this study. Researchers can further verify whether the models deliver the performance that the recognition accuracy numbers claim to deliver. The possibility of adding a battery or solar component and mounting on an unmanned aerial vehicle [10] would be an interesting future work expansion.

References

1. Chatzidimopoulos, M., Pappas, A.: Epidemiology and control of septoria pyricola in pear leaf and fruit. Journal of Plant Pathology **98**, 447–452 (11 2016). https://doi.org/10.4454/JPP.V98I3.020
2. Chollet, F.: Xception: Deep learning with depthwise separable convolutions (2017). https://doi.org/10.1109/cvpr.2017.195
3. David, R., Duke, J., Jain, A., Reddi, V.J., Jeffries, N., Li, J., Kreeger, N., Nappier, I., Natraj, M., Regev, S., Rhodes, R., Wang, T., Warden, P.: Tensorflow lite micro: Embedded machine learning on tinyml systems. CoRR **abs/2010.08678** (2020), https://arxiv.org/abs/2010.08678
4. Espressif Systems: EspressIDF: IoT Development Framework. https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/linux-macos-setup.html, Last Accessed: May 08, 2023
5. Fenu, G., Malloci, F.M.: Classification of pear leaf diseases based on ensemble convolutional neural networks **5**, 141–152. https://doi.org/10.3390/agriengineering5010009
6. Fenu, G., Malloci, F.M.: DiaMOS Plant: A Dataset for Diagnosis and Monitoring Plant Disease **11**, 2107. https://doi.org/10.3390/agronomy11112107
7. Google Inc.: TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/, Last Accessed: May 08, 2023
8. Han, H., Siebert, J.: Tinyml: A systematic review and synthesis of existing research (2022). https://doi.org/10.1109/icaiic54071.2022.9722636
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (June 2016). https://doi.org/10.1109/CVPR.2016.90

10. Karar, M.E., Alotaibi, F., Al-Rasheed, A., Reyad, O.: A Pilot Study of Smart Agricultural Irrigation using Unmanned Aerial Vehicles and IoT-Based Cloud System **10**, 131–140 (2021). https://doi.org/10.18576/isl/100115
11. Lāce, B.: Gymnosporangium species – an important issue of plant protection **71**, 95–102 (2017). https://doi.org/10.1515/prolas-2017-0017
12. Mathworks Inc.: Pretrained deep neural networks. https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html, Last Accessed: May 08, 2023
13. Mitra, A., Mohanty, S.P., Kougianos, E.: A Smart Agriculture Framework to Automatically Track the Spread of Plant Diseases Using Mask Region-Based Convolutional Neural Network. In: Proceedings of the 5th IFIP International Internet of Things Conference (IFIP-IoT). pp. 68–85 (2022). https://doi.org/10.1007/978-3-031-18872-5_5
14. Mitra, A., Mohanty, S.P., Kougianos, E.: aGROdet: A Novel Framework for Plant Disease Detection and Leaf Damage Estimation. In: Proceedings of the 5th IFIP International Internet of Things Conference (IFIP-IoT). pp. 3–22 (2022). https://doi.org/10.1007/978-3-031-18872-5_1
15. Mitra, A., Singhal, A., Mohanty, S.P., Kougianos, E., Ray, C.: eCrop: A Novel Framework for Automatic Crop Damage Estimation in Smart Agriculture. SN Comput. Sci. **3**(4), 16–pages (2022), doi: 10.1007/s42979-022-01216-8
16. Mitra, A., Vangipuram, S.L.T., Bapatla, A.K., Bathalapalli, V.K.V.V., Mohanty, S.P., Kougianos, E., Ray, C.: Everything You Wanted To Know About Smart Agriculture. CoRR **abs/2201.04754** (2022), https://arxiv.org/abs/2201.04754
17. Rehman, Z.U., Khan, M.A., Ahmed, F., Damaševičius, R., Naqvi, S.R., Nisar, W., Javed, K.: Recognizing apple leaf diseases using a novel parallel real-time processing framework based on mask rcnn and transfer learning: An application for smart agriculture **15**, 2157–2168 (2021). https://doi.org/10.1049/ipr2.12183
18. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge (2015)
19. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks (2018). https://doi.org/10.1109/cvpr.2018.00474
20. Schizas, N., Karras, A., Karras, C., Sioutas, S.: Tinyml for ultra-low power ai and large scale iot deployments: A systematic review **14**, 363. https://doi.org/10.3390/fi14120363
21. Silaparasetty, N.: Machine learning programming with tensorflow 2.0 (2020). https://doi.org/10.1007/978-1-4842-5967-2_11
22. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1409.1556
23. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision (2016). https://doi.org/10.1109/cvpr.2016.308
24. Tan, M., Le, Q.V.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: Proceedings of the 36th International Conference on Machine Learning. pp. 6105–6114 (2019)
25. Thomidis, T., Katerinis, S.: Occurrence of a fruit spot disease of pear caused by septoria pyricola in tyrnavos larissa, northern greece **98**, 845–845 (2014). https://doi.org/10.1094/pdis-09-13-0960-pdn
26. Udutalapally, V., Mohanty, S.P., Pallagani, V., Khandelwal, V.: sCrop: A Novel Device for Sustainable Automatic Disease Prediction, Crop Selection, and Irrigation in Internet-of-Agro-Things for Smart Agriculture **21**, 17525–17538 (2021). https://doi.org/10.1109/jsen.2020.3032438
27. Warden, P., Situnayake, D.: TinyML: Machine Learning with TensorflowLite on Arduino and Ultra Low Power Microcontrollers. O'Reilly Media, Inc. (2019)
28. Yang, F., Li, F., Zhang, K., Zhang, W., Li, S.: Influencing factors analysis in pear disease recognition using deep learning **14**, 1816–1828 (2021). https://doi.org/10.1007/s12083-020-01041-x
29. Zaman, F.: Tflite architecture (2020). https://doi.org/10.1007/978-1-4842-6666-3_4
30. Ziaul Haque Zim, M.: Tinyml: Analysis of xtensa lx6 microprocessor for neural network applications by esp32 soc. arXiv e-prints arXiv:2106.10652 (Jun 2021). https://doi.org/10.48550/arXiv.2106.10652