Preprints.org

Article

A Mobile App for Detecting Pests in Potato Crops

Dunia Pineda Medina , Ileana Miranda Cabrera , Rolisbel Alfonso De la Cruz , Lizandra Guerra Arzuaga , Sandra Cuello Portal , Monica Bianchini [*]

Article

A Mobile App for Detecting Pests in Potato Crops

Dunia Pineda Medina [1], Ileana Miranda Cabrera [1,2], Rolisbel Alfonso de la Cruz [2],
Lizandra Guerra Arzuaga [2], Sandra Cuello Portal [2] and Monica Bianchini [3,*]

[1]   Universidad de la Habana
[2]   Centro Nacional de Sanidad Agropecuaria
[3]   Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche, Università degli Studi di Siena
*   Correspondence: monica.bianchini@unisi.it

Abstract: Artificial intelligence techniques are now widely used in various agricultural applications, including the detection of devastating diseases such as late blight (*Phytophthora infestans*) and early blight (*Alternaria solani*) affecting potato (*Solanum Tuberorsum L.*) crops. In this paper, we present a mobile application for potato crop pest detection based on deep neural networks. The images were taken from the PlantVillage dataset with a batch of 1000 images for each of the three identified classes. An exploratory analysis of the architectures used for early and late blight diagnosis in potatoes was performed, achieving an accuracy of 98.76%, with MobileNetv2. Based on the results obtained, an offline mobile application was implemented, supported on devices with Android 4.1 or later, also featuring an information section on the 27 pests affecting potato crops and a gallery of symptoms. For future work, segmentation techniques will be used to highlight the damaged region in the potato leaf by evaluating its extent and possibly identifying different types of pests affecting the same plant.

Keywords: deep neural networks; image processing; classification of potato crop pests

---

1. Introduction

The automatic detection of pathogens in plants, as early as possible and without damaging the plant, is an approach that is enjoying increasing success in the agri–food sector. In automatic detection, the basic assumption is that a diseased plant looks different from a healthy one. For example, leaves can exhibit subtle color differences, often invisible to the human eye but which can be captured using techniques such as spectral imaging. Pest detection is often complicated because pests and their eggs are often found under the canopy of plants and are therefore difficult to detect. They are often very small and show a very local distribution. Crops in general could be affected by multiple pests at the same time. Therefore, not only high–resolution detection, but also local and organism–specific detection is needed. High–resolution imaging, combined with deep learning techniques, especially Convolutional Neural Networks (CNNs), could have the potential for precision agriculture for standard and greenhouse crops. In both cases, large quantities of labeled images from different situations (locations, seasons, crop varieties) are needed to sufficiently train deep learning algorithms. Moreover, augmentation and smarter training techniques are necessary to overcome the lack of real data and labeled images. Transfer learning has also been useful in the detection and diagnosis of diseases in agricultural crops, without ignoring the multiplicity of applications it has [1].

Particularly, potato (*Solanum tuberosum L.*) crops are constantly affected by the incidence of parasites which cause a decrease in their yield every year. Being a widespread crop in the world, the control of its production requires attention, and the problem of automatic disease recognition from leaf images via CNNs has been the subject of much recent literature, such as [2–7], to cite only some contributions. For instance, in [8], potato tuber diseases were diagnosed using the VGG architecture by adding new dropout layers to avoid overfitting. As a result, 96% of the test images were classified correctly. After comparing MobileNet, VGG16, InceptionResNetV2, InceptionV3, ResNet50, VGG19 and Xception architectures, in [9], it was found that VGG16 had the highest accuracy (99.43%) on test data for the diagnosis of late blight and early blight, the most incident diseases in potato crops. Finally, in [10], a novel hybrid deep learning model, called PLDPNet, has been proposed, for automatic segmentation

and classification of potato leaf diseases. The PLDPNet utilizes auto–segmentation and deep feature ensemble fusion modules to enhance disease prediction accuracy, with an end–to–end performance of 98.66% on the PlantVillage dataset ([11], https://www.kaggle.com/datasets/emmarex/plantdisease).

The versatility of CNNs allows their implementation from different platforms, including mobile devices. Mobile applications achieved rapid popularity because, in addition to being practical and lightweight, they simplify access to information and promote their widespread use. Their ecosystem is made up of several factors: infrastructure, operating system (OS), information distribution channels, etc. Nowadays, almost everyone owns a smartphone, whether it is an Android, iOS or other operating system device. Despite their diffusion, in Cuba a large part of the population can only afford phones with low performance (Android version over 4.0, 2G data network with a population coverage of 85%, internal memory 1 GB, etc.). The price–quality ratio is an obstacle to technological updating and, therefore, obtaining a practical tool, low demanding in term of computational resources, for the detection of potato pests is a necessary strategy for closing surveillance gaps in crop campaigns threatened by more than one disease.

The objective of this paper is to release an offline mobile application with the most effective machine learning architecture for the diagnosis of fungal blight in potatoes. The mobile application developed is compatible with Android versions higher than 4.1, has a storage capacity of 77.57 MB, and does not require Internet connection or mobile coverage. Other similar proposals can be found in literature, as in [12], where a mobile app based on the MobileNetv2 architecture was developed, able to classify five pest categories: general early blight, severe early blight, severe late blight, severe late blight fungus and general late blight fungus. The model achieved an accuracy of 97.73%. Nonetheless, this study, as well as [13–15], requires high resolution images and/or advanced features in the technological infrastructure, incompatible with the characteristics of Cuban mobiles, which are mostly on the way to obsolescence and unable to take even medium quality pictures. In [16], a mobile application called VegeCare was devised for the diagnosis of diseases in potatoes, yielding a 96% accuracy. However, it is a proprietary software, difficult to access for the Cuban community. Moroever, most of these studies propose an architecture that requires connection to an external server for image processing, as in [14]. While it is true that this works without extenuating circumstances in many countries, due to the availability of resources and access to free online platforms, it must be considered that in Cuba there are still planting regions where there is no mobile coverage or very low signal, which would hinder access to available international solutions.

Many mobile apps for smart agriculture have recently been devised based on deep learning [17], sometimes founded on proprietary software. However, these apps, in addition to not being free of cost, can only be installed on devices with current Android versions and, normally, refer to a client–server architecture where the information is stored in external databases. Therefore, they require mobile networks and an external server with a MySQL manager for queries [14]. In Cuba, the company GeoCuba has focused its efforts on image processing in the agricultural sector, mainly for the control of sugar cane and rice cultivation. Using satellite photos, drones and AI techniques, damages in these crops can be identified; however, this requires advanced tools to capture images in real time and platforms with high computational performance. Not to mention that the distance at which the images are taken may hinder the efficiency of the diagnosis.

All the above motivations push towards having a simple mobile app, which in addition of being free, offline and suitable for the characteristics of devices with low performance, can also cover the role of a decision assistant. The real–time diagnosis of the main pests contributes to the reduction of the risk of crop losses, to the early identification of the type of parasite, to the reduction of the use of pesticides and, therefore, to ecological sustainability. It includes an important strategic component as it is an informative tool that helps non–expert personnel to know about different diseases present in the crops, produced by insects, viruses, bacteria and nematodes, as each one has degenerative factors on a medium or large scale in the potato cultivation.

The rest of the paper is organized as follows. In the following section, the PlantVillage dataset and the experimental setting are described. In Section 3, our experimental results are reported, assessing the superiority of the MobelNetv2 architecture—as a good compromise between computational lightness and performance—to be included in a mobile app for potato pest detection. The PPC (Potato Pest Control) app is briefly described in the subsequent Section 4. Finally, Section 5 traces some conclusions and future perspectives.

2. Materials and Methods

2.1. The PlantVillage Dataset

Potato crop leaf images were used as a case study, with a focus on the most incident diseases, late blight and early blight, identifying three classes by including healthy leaves. Late blight (Phytophthora infestans) is a polycyclic disease, which develops rapidly at moderate temperature and high humidities, so it can collapse the crop in less than a week. Symptoms occur primarily on plant leaves and tuber. Young leaves are more susceptible to infection, which starts with light–colored spots and turns dark brown on the leaf surface [18]. Early blight (Alternaria solani) forms very similar lesions on both leaves and stems and can affect the entire plant. On the leaves the symptoms manifest themselves with light brown spots which, as the disease progresses, become dark, presenting concentric rings or being limited by leaf veins [18]. In both pests the main control measure suggested is to correctly identify the problem through early disease classification. The data used in this study are extracted from the PlantVillage dataset (https://www.kaggle.com/datasets/emmarex/plantdisease), taking the images with magnification. An average of 1000 images were used for each class, with resolution 96×96 pixels per inch, dimensions 256×256 pixels, and 24 bits in depth (see Figure 1). Seven hundred training images were taken, 200 for validation and 100 for model evaluation for each class.
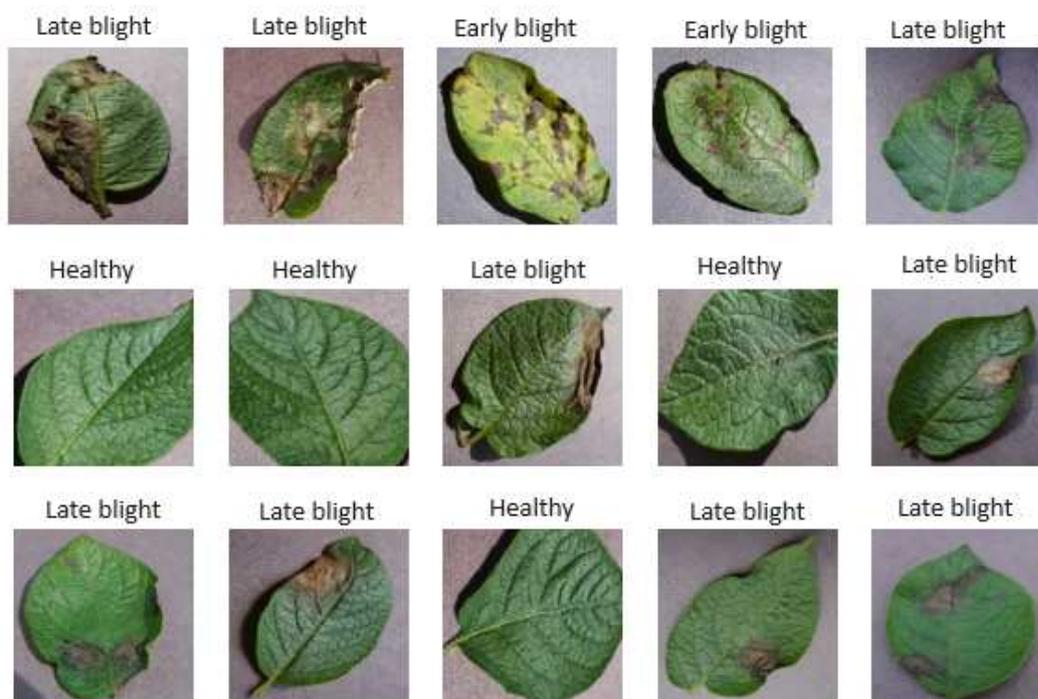
Figure 1. Sample images taken from the PlantVillage dataset of potato leaves corresponding to late blight, early blight and healthy leaves.

2.2. Experimental Setting

The experiments were carried out with a Windows 11 Pro operating system, on a x64 processor, Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz and with 8GB RAM. For image processing, we made use of the Anaconda Navigator platform (v.2.1.4) and the Spyder IDE (v.5.1.5), and the TensorFlow (v.2.10.1), Keras (v.1.1.2), Matplotlib (v.3.7.1) and MumPy (v.1.23.4) libraries. Python and the Python interpreter (v.3.9) were used for the software implementation. The Android Studio development environment, the Kotlin programming language and TensorFlow–lite dependencies were used to develop the mobile application.

Five widely used CNNs have been evaluated, namely MobilNetv2 [19], VGG16 [20], VGG19 [21], InceptionV3 [22] and Xception [23], calculating the accuracy of each model, to select the best performing one. For each of the analyzed architectures, the hyperparameters were set as listed below (with values in common for the first iterations), in order to make a fair comparison between models.

- Number of epochs: 10, 50
- Activation function for the output layer: Softmax
- Optimizer: Adam
- Loss function: Sparse Categorical Cross–entropy
- Batch size: 32
- Metrics: Accuracy

The first considered model was MobileNetv2, which is a lightweight architecture particularly tailored to mobile applications, whose computational cost and processing time are significantly lower than the rest of the architectures tested in our experiments. It is based on an inverted residual structure and, as a whole, contains the initial full convolutional layer with 32 filters, followed by 19 residual bottleneck layers. ReLU6 is used as the activation function because of its robustness when used with low–precision hardware. Finally, the network has an image input size of 224×224 [19]. Instead, the VGG16 model has 13 convolutional layers followed by 13 fully connected layers—with ReLU activation—only 16 of which have learnable weights (hence the name) [20]. The network has an image input size of 224×224. VGG19 shares the same structure as VGG16, with the addition of three convolutional layers—thus it has 19 trainable layers [21]. Inceptionv3 has a 42–layer architecture and processes images of size 229×229. It is computationally less expensive with respect to previous Inception architectures (v1 and v2) and can easily be re–trained for custom image classification problems [22]. Finally, Xception is an extension of the Inception model, which uses the standard Inception modules with depth–separable convolutions. The Xception architecture has 36 convolutional layers forming the feature extraction base of the network. The 36 convolutional layers are structured into 14 modules, which have linear residual connections, except for the first and last modules. In short, the Xception architecture is a linear stack of depth–wise separable convolutional layers with residual connections. The convolutional and separable convolutional layers are followed by batch normalization [23]. All experiments described in the following were based on pretrained architectures [1], fine–tuned on the PlantVillage dataset. Three steps were carried out:

- **Step 1 –** Firstly, the models were trained for ten epochs.
- **Step 2 –** Then, three new layers were added: (i) a dropout layer with a rate of 0.3 to avoid overfitting, (ii) a dense layer with ReLU activation functions and (iii) a softmax activation function in the output layer.
- **Step 3 –** Finally, the new incorporated layers were kept and the number of epochs was increased to 50.

---

[1] The CNN models presented in this section are saved in Github and are freely downloadable at https://github.com/dkpineda88/TransferLearninPapas.git.

In this work, the trained models were converted into tflite files for optimization and processing in the Android Studio platform. Finally, a model deployment module was built to store the trained neural networks in the Kotlin framework [2].

3. Experimental Results

The increase in the cost of energy and raw materials in Cuba is causing a new concept in agricultural production techniques, and the use of IT tools, mainly based on artificial intelligence, paves the way for developments capable of revolutionizing agricultural work. The goal of smart agriculture is to increase profits and, of course, reduce the risks of capital loss and destruction of natural resources. Mobile applications for disease detection represent a smart strategy and their use is currently essential to strengthen food sustainability, especially given the lack of investment in infrastructure plaguing the Cuban agricultural system. To obtain a CNN model capable of significantly reducing computational costs, being adaptable to the performance of mobile devices and capable of processing images effectively, it is necessary to take into account several incident factors (e.g., the number of model parameters and processing times), mainly related to the limited computational resources available. In fact, traditional deep learning models cannot be applied directly to mobile devices.

Therefore, after investigating lightweight neural network architectures and using transfer learning to limit the computational load due to training, the MobileNetv2 architecture was found to have the best adaptability to the data, with the highest level of accuracy, the lowest number of parameters and the lowest number of epochs (see Table 1).

Table 1. Hyperparameter setting and best results for the five CNN models.

CNN type	Step [1]	# trained param.	# frozen param.	# epochs	Accuracy	Loss	Model size
MobileNetv2	2	64,323	3,540,265	10	0.9876	0.036	3.89 MB
VGG16	2	1,539	14,714,688	10	0.94	0.38	14.2 MB
VGG19	1	1,539	20,024,384	10	0.9844	0.0415	19.2 MB
Inceptionv3	1	153,603	21,802,784	10	0.91	5.7269	21.4 MB
Xception	3	6147	20,861,480	50	0.9467	0.1394	21.1 MB

[1] The **Step** column describes at which of the above described training steps the network reaches its best performance.

Indeed, after training the MobileNetv2 only for ten epochs, data overfitting was observed. Instead, adding the layers described in **Step 2**, the accuracy on the validation set remains relatively aligned with that on the training data (see Figure 2). In other words, based on the test set, the predicted values are close to the observed values (Figure 3).

These results differ from those obtained in [6] where, after applying ten deep learning models such as DenseNet201, DenseNet121, NasNetLarge, Xception, ResNet152v2, EfficientNetB5, EfficientNetB7, VGG19 and MobileNetv2 along with the hybrid model EfficientNetB7–ResNet152v2 for classification, it resulted that DenseNet201 obtained the highest accuracy, equal to 98.67%, with a validation error of 0.04. However, the model covers not only potato pests but also tomato and bell pepper pests, with a total of 15 disease classes. Instead, in [5] the VGG16 model was selected, achieving 100% accuracy on the test data, after also evaluating VGG19, MobileNetv2, Inceptionv3 and Resnet50v2. Anyway, neither network size nor processing speed were taken into account in this study, although they are necessary elements for a model to be encapsulated in a mobile app, which is the ultimate goal of the present research.

---

[2] For this purpose, the following dependencies were installed: 'org.tensorflow:tensorflow-lite:2.4.0','org.tensorflow:tensorflow-lite-support:0.1.0','org 'org.tensorflow:tensorflow-lite-gpu:2.3.0'.

Figure 2. MobileNetv2 accuracy and loss on the training and validation data, respectively.
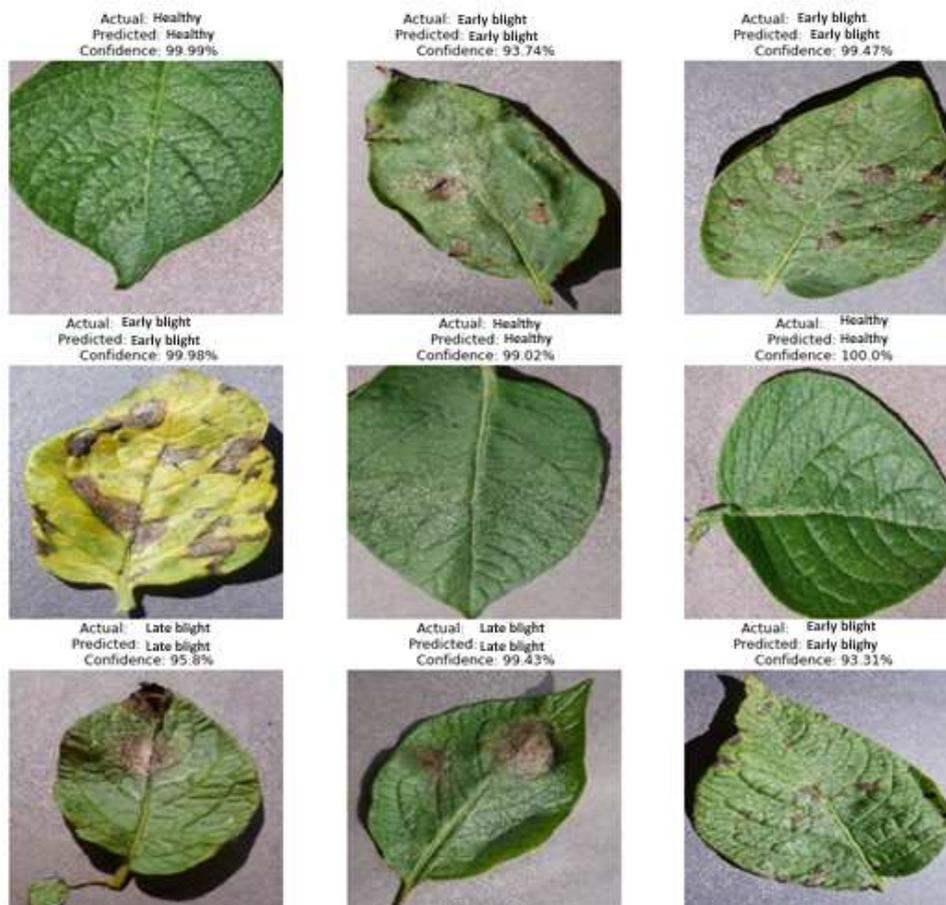

Figure 3. MobileNetv2 prediction on test data. The confidence with which the network takes its decision is reported for each sample image.

4. Deployment of the PPC Mobile App

The PPC mobile app is compatible with Android version higher than 4.1, requires 77.57 MB of storage and does not require Internet connection or mobile coverage. The first interface of the application presents a brief description of the project (see Figure 4a) and a side menu with the following options: (i) Home (link to the main page of the application), (ii) Crop pests, (iii) Diagnosis, and (iv) Symptom images (Figure 4b).


(a)　　　　　　　　　　　　(b)
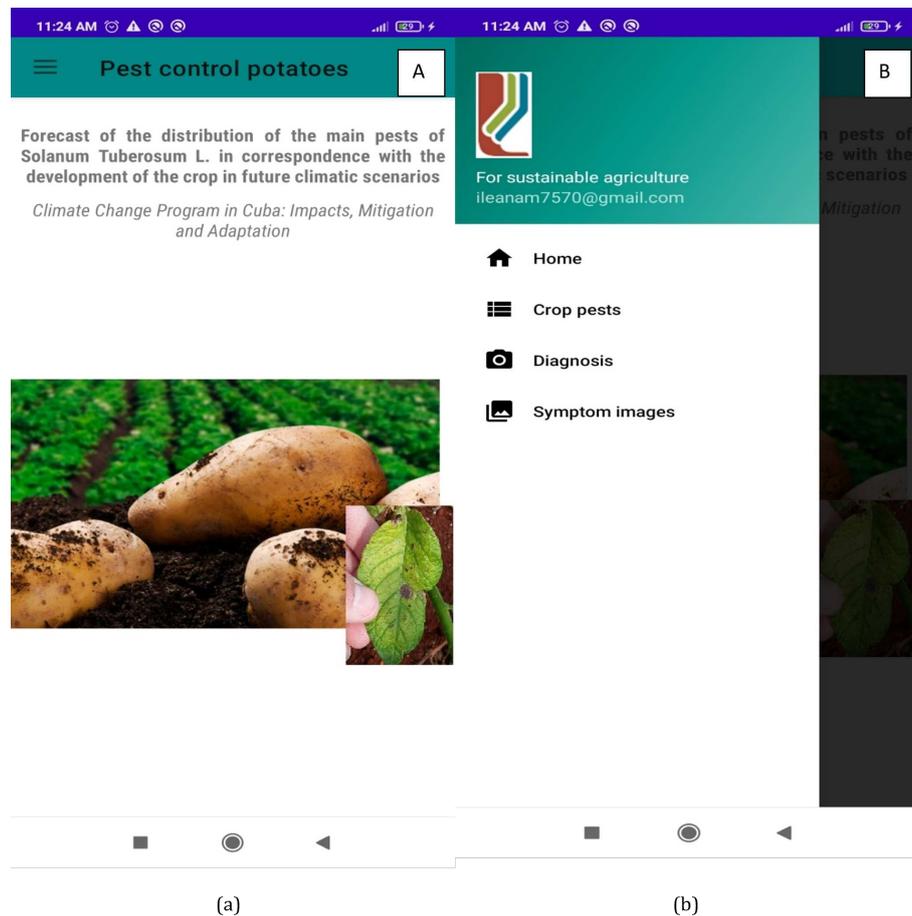
Figure 4. Graphic interface of the PPC app main page (**a**) and side menu (**b**).

Option (ii), has the objective of showing the 27 pests that affect potato crops and, at the same time, briefly describes their main characteristics: scientific name, symptomatology, epidemiology and cycle and control techniques. The first interface of this section shows the list of diseases subdivided by the causal agents of the disease (Fungi, Bacteria and Viruses, Insects and Nematodes, Figure 5a). Clicking on the disease of interest will display its description (Figure 5b).

(a)   (b)
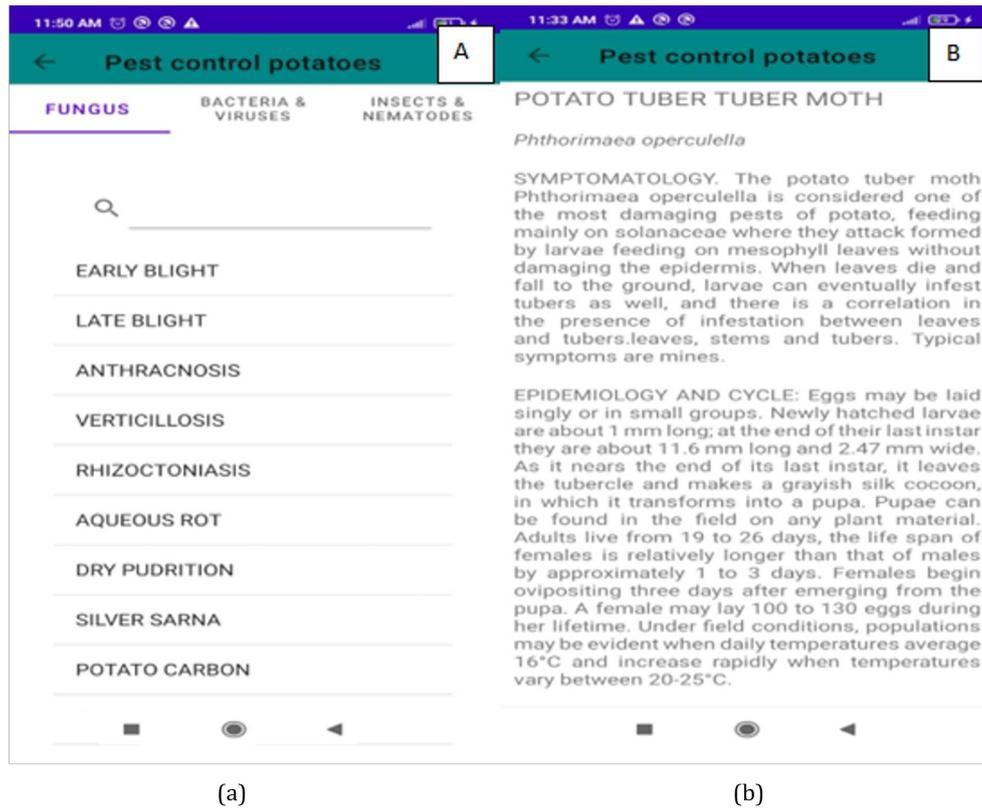
Figure 5. Graphic interface of the PPC app for the list of pests (**a**) and the detailed description of a specific disease (**b**).

Option (iii) responds to the main objective of this work as it is responsible for diagnosing, through an image, the percentage of presence of late blight, early blight or if the plant is healthy (Figure 6a). Images can be selected from the gallery of the mobile device or taken in real time in the field. Finally, option (iv) allows the user to visualize, through images, the behavior of the symptoms according to the diseases described in option (ii) (Figure 6b).

(a)   (b)
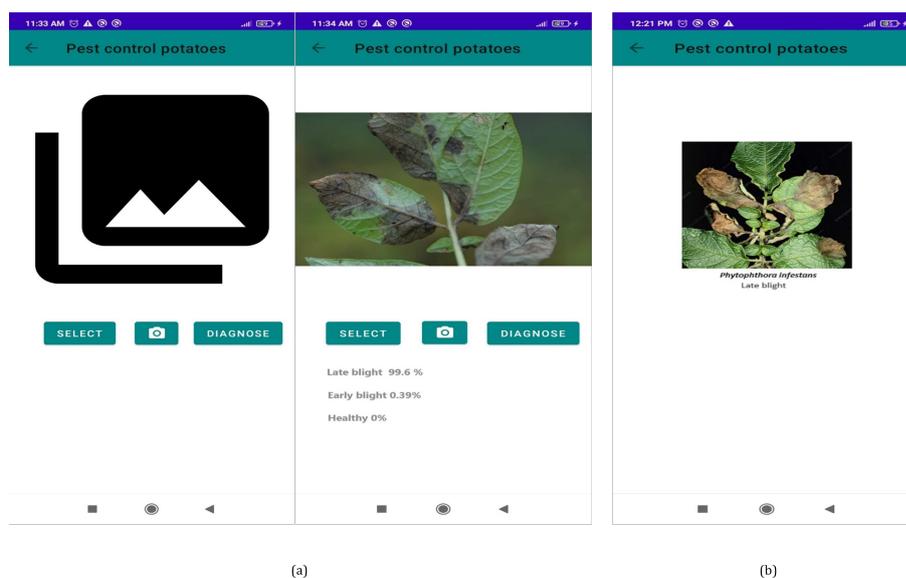
Figure 6. Graphic interface of the PPC app for the diagnosis of late blight and early blight given an image (**a**); a diseased leaf example (**b**).

5. Conclusions

In this paper, we have proposed an experimental study on different deep network architectures in order to find the most suitable to be used in a mobile app for potato pest identification. A major constraint was that of choosing a lightweight model to be used on obsolete hardware/software mobile phones in Cuba, also unable to access the network. Preliminary experimental results are promising. Future work will be devoted to preventively apply some segmentation techniques on the leaf images to diagnose not only the type of pest but also its severity (namely the extension of the leaf surface interested by the disease), which is important especially when leaves may be affected by more than one disease. Moreover, enriching the image collection by the final users —who can capture pictures with the cellular phone in real conditions—will be valuable, especially in view of a future where cloud computing will be an option also in Cuba.

References

1. Yu, F.; Xiu, X.; Li, Y. A Survey on Deep Transfer Learning and Beyond. *Mathematics* **2022**, *10*, 3619.
2. Agarwal, M.; Sinha, A.; Gupta, S.K.; Mishra, D.; Mishra, R. Potato Crop Disease Classification Using Convolutional Neural Network. In *Smart Systems and IoT: Innovations in Computing. Smart Innovation, Systems and Technologies*, 2020; Volume 141.
3. Hasi, J.M.; Rahman, M.O. Potato Disease Detection Using Convolutional Neural Network: A Web Based Solution. In *Machine Intelligence and Emerging Technologies—Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, **2023**; Volume 490.
4. Kang, F.; Li, J.; Wang, C.; Wang, F. A Lightweight Neural Network-Based Method for Identifying Early–Blight and Late–Blight Leaves of Potato. *Appl. Sci.* **2023**, *13*, 1487.
5. Krishnakumar, B.; Kousalya, K.; Indhu Prakash, K.V.; Jhansi Ida, S.; Ravichandra, B.; Rajeshkumar, G: Comparative Analysis of Various Models for Potato Leaf Disease Classification using Deep Learning. In Proceedings of 2023 Second International Conference on Electronics and Renewable Systems (ICEARS), 2023; pp. 1186–1193.
6. Kumar, Y.; Singh, R.; Moudgil, M.R.; Kamini, G. A Systematic Review of Different Categories of Plant Disease Detection Using Deep Learning–Based Approaches. *Arch. Comput. Methods Eng.* **2023**, *30*, 4757–4779.
7. Sharma, R.; Singh, A.; Attri, K.; Jhanjhi, N.Z.; Masud, M.; Jaha, E.S.; Sk, S. Plant Disease Diagnosis and Image Classification Using Deep Learning. *Comput. Mater. Contin.* **2022**, *71*, 2125–2140.
8. Oppenheim, D.; Shani, G. Potato Disease Classification Using Convolution Neural Networks. *Adv. Anim. Biosci.* **2017**, *8*, 244–249.
9. Islam, F.; Hoq, M.N.; Rahman, C.M. Application of Transfer Learning to Detect Potato Disease from Leaf Image. In Proceedings of the 2019 IEEE International Conference on Robotics, Automation, Artificial–intelligence and Internet–of–Things (RAAICON), 2019; pp. 127–130.
10. Arshad, F.; Mateen, M.; Hayat, S.; Wardah, M.; Al-Huda, Z.; Gu, H.Y.; Al-antari, M.A. PLDPNet: End–to–end hybrid deep learning framework for potato leaf disease prediction. *Alex. Eng. J.* **2023**, *78*, 406–418.

11. Lee, T.-Y.; Yu, J.-Y.; Chang, Y.-C.; Yang, J.-M. Health detection for potato leaf with convolutional neural network. In Proceedings of the 2020 Indo—Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN), 2020; pp. 289–293.

12. Chen, W.; Chen, J.; Zeb, A.; Yang, S.; Zhang, D. Mobile convolution neural network for the recognition of potato leaf disease images. *Multimedia Tools Appl.* **2022**, *81*, 20797–20816.

13. Chen, J.-W.; Lin, W.-J.; Cheng, H.-J.; Hung, C.-L.; Lin, C.-Y.; Chen, S.-P. A Smartphone–Based Application for Scale Pest Detection Using Multiple–Object Detection Methods. *Electronics* **2021**, *10*, 372.

14. Karar, M.E.; Alsunaydi, F.; Albusaymi, S.; Alotaibi, S. A new mobile application of agricultural pests recognition using deep learning in cloud computing system. *Alex. Eng. J.* **2021**, *60*, 4423–4432.

15. Wang, F.; Wang, R.; Xie, C.; Zhang, J.; Li, R.; Liu, L. Convolutional neural network based automatic pest monitoring system using hand–held mobile image analysis towards non–site–specific wild enviroment. *Comput. Electron. Agric.* **2021**, *187*.

16. Ruedeeniraman, N.; Ikeda, M.; Barolli, L. Performance Evaluation of VegeCare Tool for Potato Disease Classification. In *Advances in Networked–Based Information Systems – NBiS 2020*, *Advances in Intelligent Systems and Computing*, Springer, 2021; Volume 1264.

17. Altalak, M.; Ammad Uddin, M.; Alajmi, A.; Rizg, A. Smart Agriculture Applications Using Deep Learning Technologies: A Survey. *Appl. Sci.* **2022**, *12*, 5919.

18. Dong, S.-M.; Zhou, S.-Q. Potato late blight caused by Phytophthora infestans: From molecular interactions to integrated management strategies. *J. Integr. Agric.* **2022**, *21*, 3456–3466.

19. Sandler, M; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018; pp. 4510–4520.

20. Liu, S.; Deng, W. Very deep convolutional neural network based image classification using small training sample size. In Proceedings of 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 2015; pp. 730–734.

21. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large–Scale Image Recognition, *CoRR abs/1409.1556*, **2014**.

22. Szegedy, c.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016; pp. 2818–2826.

23. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017; pp. 1800–1807.