Virtual Laboratory for Identifying Fruit Flies using Deep Learning and Augmented Reality on Mobile Devices

DIEGO FALCONÍ[1,*], ACKSOHN MAQUILON[1], JOSE FALCONÍ[2]
[1]Department of Computer Science, Software Engineering Program,
[1] Technical University of Cotopaxi,
ECUADOR

[2]Department of Computer Science, Software Engineering Program,,
Armed Forces University ESPE,
Latacunga, Cotopaxi,
ECUADOR

*Corresponding Author

Abstract: - Agronomy students need to identify agricultural pests like fruit flies of the genus Anastrepha, a task often hindered by the lack of interactive tools. This study develops a mobile "virtual laboratory" that integrates augmented reality (AR) with deep learning for species recognition in biointensive fruit-growing ecosystems. The MobileNetV2 architecture was chosen over VGG16 for its greater efficiency-to-accuracy ratio. The Barracuda inference worked with the Unity engine so that it could handle data in real time. A test set of 200 images was used for experimentation, yielding an F1 score of 0.98 and an accuracy of 98%. Usability tests were also conducted to measure the optimal AR scanning distance, which showed that the ideal distance was 15cm. Therefore, it can be concluded that this is an effective tool for species identification, with low latency and no internet connection required for cloud services.

Key-Words: - Augmented Reality, MobileNetV2, Virtual Laboratory, Unity, Barracuda, ONNX.

Received: June 29, 2025. Revised: November 21, 2025. Accepted: March 7, 2026. Published: June 2, 2026.

1 Introduction

Precision agriculture and biointensive fruit growing are paradigms for optimizing yields and minimizing environmental impact. However, the application of these sustainable methods in the Andes, specifically in the Cotopaxi province of Ecuador, often encounters the problem of fly pests that are not easily identifiable. Amon then, "fruit flies of the genus Anastrepha (Diptera: Tephritidae) represent a significant phytosanitary pest, as they cause direct damage to the fruit pulp and economic losses for local producers", [1]. The correct taxonomic identification of these species is, therefore, fundamental for effective integrated pest management strategies. Agronomy students and field workers find it challenging to accurately classify the subtle physical characteristics required (ovipositor structures, wing venation patterns, etc.), [2].

The traditional teaching approach, represented by dichotomous keys, fixed two-dimensional charts, and preserved specimens, might not allow for meaningful interaction in order to be meaningful, [2]. In rural schools, the gap between what is known in the books and what students are actually skilled in field knowledge still exists, as students do not have access to an entomology laboratory with the required tools.

For this reason, AR-AI integration is proposed as a solution for opening a novel approach to solve this issue. Augmented reality (AR) enables students to manipulate virtual 3D samples from multiple viewpoints by overlaying digital content onto the real world, supporting situated learning, [3]. Meanwhile, deep learning automates the image recognition step. Despite studies successfully classifying pests using neural networks, they remain largely based on cloud architectures, which are unfeasible in poor network regions, [4].

The core goal of this study is to develop a mobile "Virtual Laboratory" applying Augmented Reality to train morphological features and a Convolutional Neural Network (CNN) to automatically detect images offline. We apply the MobileNetV2 architecture and the Unity Barracuda inference engine to develop a state-of-the-art, real-time training tool that is embedded right on the mobile device (edge computing), allowing access to cutting-edge diagnostic equipment for agricultural engineering students.

1.1 Mobile Computing and the Android Ecosystem

Mobile devices were first made to help people talk to each other, but now they are sophisticated computer platforms that can be used in science. The Android ecosystem is the most popular in rural Ecuador since it is open source and works with a lot of different devices. Because so many people use Android, it's the best platform for easily available instructional resources, [5]. This proposed algorithm utilizes hardware peculiarities of modern smartphones, such as their high-resolution digital cameras for taking pictures and their specialized GPUs (Graphics Processing Units) for running neural inference techniques and making 3D images.

1.2 3D Modelling and Texturing

The virtual specimens must have a photorealistic model to ensure the application's educational value. For polygonal modelling, Blender was utilized due to its advanced sculpting and UV mapping capabilities. To represent the head, thorax, and abdominal components, a polygonal mesh based on spherical primitives was created. Importantly, semi-transparent textures mapped from microscopic photographs were used to mimic the wing patterns, which are essential for distinguishing Anastrepha species. Lastly, as illustrated in Figure 1, a skeletal rigging system was used to replicate natural anatomical positions.

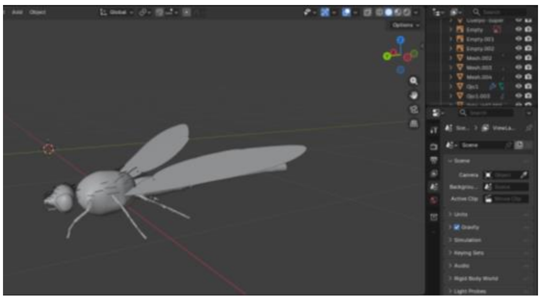

Fig. 1: The skeletal structure of the Anastrepha specimen was modelled using Blender
Source: created by the authors

1.3 Augmented Reality (AR)

Augmented reality (AR) involves adding digital information to the real world in real time. Unlike virtual reality (VR), which creates a separate environment, augmented reality (AR) enhances the user's understanding of their current surroundings. AR addresses the limitations of 2D media in entomology education by enabling the visualization of volumetric data. For this project, we used the Vuforia Engine, a Software Development Kit (SDK) that uses computer vision to monitor planar images (Image Targets). The technique allows the student to virtually inspect the insect's morphology with a simple movement of the device by recognizing a certain QR pattern and associating a high-fidelity 3D model with it, as illustrated in Figure 2.

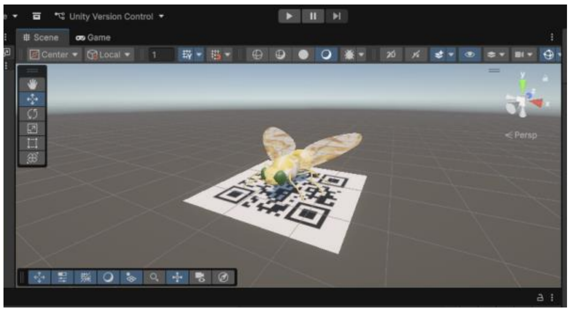

Fig. 2: Using the Vuforia Engine to Connect 3D Models with Physical QR Codes
Source: created by the authors

1.4 Deep Learning and Convolutional Neural Networks (CNN)

Deep learning is now the norm for image classification. Convolutional Neural Networks (CNNs) are particularly promising since they can automatically learn hierarchical feature representations from raw pixel data. Nevertheless, deployment of CNNs on mobile devices is often restricted by memory size and latency. To overcome this, we employed the MobileNetV2 architecture. This model applies depth-wise separable convolutions, which dramatically reduce computational cost (Mult-Adds) and the number of parameters while still maintaining accuracy compared to traditional networks. We applied transfer learning to our fruit fly dataset using a pre-trained ImageNet model (see Figure 3 for the altered layer structure).

```
# ==============================
# 5. Construir el modelo con MobileNetV2
# ==============================
base_model = MobileNetV2(weights='imagenet', include_top=False, input_shape=(128, 128, 3))

# Congelar capas del modelo base
for layer in base_model.layers:
    layer.trainable = False

# Agregar capas personalizadas
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(128, activation='relu')(x)
predictions = Dense(1, activation='sigmoid')(x)

model = Model(inputs=base_model.input, outputs=predictions)

model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

Fig. 3: Custom layers added to the MobileNetV2 architecture
Source: created by the authors

1.5 Edge Computing with Unity Barracuda

Traditional mobile artificial intelligence applications typically move tasks to the cloud, which causes lag and necessitates internet access. In order to enable "Edge AI" to process data locally on the device, we integrated the Unity Barracuda inference engine. Barracuda connects the mobile hardware to the learned neural network. An interoperability standard called the Open Neural Network Exchange (ONNX) format, [6] is used to export the trained model from the Python environment. The pipeline in Figure 4 illustrates how the application may carry out inference in real-time directly on the phone's CPU or GPU after importing this file into Unity.

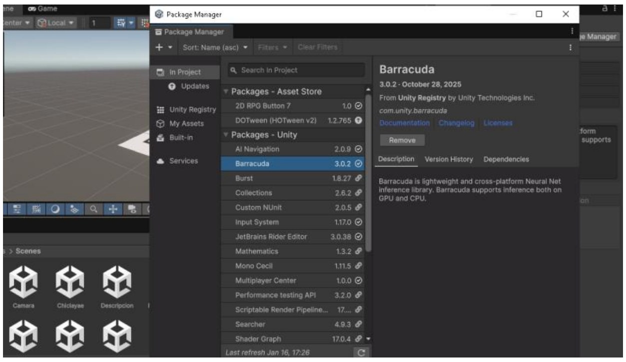

Fig. 4: Integration of the trained ONNX model into the Unity environment
Source: created by the authors

2 Methodology

The proposed system architecture was developed using a modular pipeline. This included edge-deployment integration, training optimization, neural network mathematical modelling, and data collection.

2.1 Dataset Construction and Mathematical Augmentation

To ensure the model's ability to generalize across different environments, a large dataset of 5,000 high-resolution images was created. The dataset's positive class, Anastrepha spp. The negative class, encompassing non-target insects, background noise, and foliar elements, was meticulously balanced. As building large biological datasets has its own inherent limitations, an online data augmentation method was used in the training process. The second methodology reduces the modelling overfitting to certain orientations or backgrounds by simulating the variability in the field conditions, such as camera rotation, varying scale, changes in illumination, etc.

In formal usage, the augmentation was defined as a set of affine transformations performed on an input image tensor. In addition, a transformation matrix M, incorporating rotation, scaling, and translation operations, can be obtained with equation (1), to use to obtain the transformation coordinates for a pixel initially located at coordinates such as (x, y), [7].

$$[x', y', 1]^T = M \cdot [x, y, 1]^T \quad (1)$$

The matrix M, which contains translation vectors $t_{x,y}$, scaling factors $s_{x,y} \in [0.8, 1.2]$, and rotation parameters $\theta \in [-30°, 30°]$, thus makes for a stochastic transformation process. This process requires the convolutional neural network (CNN) to determine rotation-invariant features, which is important for insect detection in unstructured field situations where the subject's orientation by camera frame varies, [8]. In addition, the horizontal and vertical orientation random flips were employed, with a probability of 0.5, to control for bilateral symmetry.

2.2 3D Modeling and Visualization

Using Blender, high-fidelity 3D models were made to make it easier to accurately identify taxonomic groups in the Augmented Reality (AR) environment. The modelling pipeline put topological optimization first while still making sure that the biology was accurate for mobile rendering. This was done in two steps: first, high-polygon models were sculpted to show fine details, and then retopology was done to make low-polygon assets that worked well for real-time rendering.

Using planar mapping and high-resolution texture maps, we were able to add important taxonomic features, such as the unique wing venation patterns (the S-band and V-band, in particular), [9]. This method makes sure that the visual diagnostics that entomologists use to tell Anastrepha apart from other Tephritidae species will still work. To make the AR environment look more like real insects moving, a skeletal rigging system was added. This system gives bones and skin weights a hierarchy. This got users more involved and helped them learn.

2.3 Mathematical Formulation of the CNN Architecture

It has been opted to consider the MobileNetV2 architecture as the basis for the classification task, due to the limited computation capacity of mobile devices, especially in terms of their small battery life and thermal throttling. MobileNetV2 uses a depthwise separable convolution, whereas CNNs based on regular convolutions (e.g., VGG16) are more expensive to compute. This new architecture separates a regular convolution into two layers: a pointwise 1x1 convolution to combine features, and a depth-wise convolution for space filtration, [10].

This decomposition greatly reduces the number of math operations. For cheap computing (2), the percentage of η needed for saving the cost of computation is represented as η:

$$\eta = (1 / N) + (1 / D_K^2) \tag{2}$$

Where the DK is the dimensionality of the spatial kernel, and N is the number of output channels. A straightforward 3x3 kernel (DK = 3) is the cost of computation that is typically about 8-9-fold cheaper compared to conventional convolutions, [11]. This is even better in the context of inverted residuals in linear bottlenecks, as the network keeps the information in low-dimensional bottlenecks and expands to high-dimensional ones for non-linear transformations. This accelerates the flow of information throughout the network.

2.4 Comparative Architecture Analysis

We extensively compared with other state-of-the-art architectures widely used in computer vision to confirm the choice of MobileNetV2. We compared model complexity (parameter count), physical storage capacity, and computation load (FLOPs) in order to determine the trade-offs for download times and the size of application bundles. These metrics are presented in Table 1.

Table 1. Computational Complexity and Efficiency Comparison

Architecture	VGG16	ResNet50	MobileNetV2
Parameters (Millions)	138.3 M	25.6 M	3.4 M
FLOPs (Millions)	~15,300	~3,800	~300
Model Size (MB)	~528	~98	~14
Edge Suitability	Low (High Latency)	Medium (Moderate Lag)	High (Real-Time)

Source: created by the authors

It was observed that VGG16 has many parameters (138M), and this would make the model much larger than 500MB. This also means that it is not suitable for mobile and would overfit on small datasets. ResNet50 is a decent middle ground, but on mid-sized machines, its processing load of approximately 3.8 GFLOPs can still give noticeable latency points. MobileNetV2 itself has just 3.4 million parameters and about 300 million floating-point operations per second (MFLOPs), but it achieves good accuracy and is well balanced. The point here is to say that it can predict on the edge constantly and never fail. This quantitative approach (accuracy and computation cost) is consistent with standard algorithm selection procedures in many of the investigated areas, [12].

2.5 Setting up and Improving the Training Process

Then it was trained on the ImageNet dataset using Transfer Learning, which started the network with weights trained on the ImageNet dataset. This initialization helps the model leverage learned low-level features (such as those with simple edges and textures) and achieve the goal of insect categorization much more quickly.

This means that it has swapped out the last fully connected categorization layer for one able to handle binary output, such as what our task does. The optimization was to reduce Binary Cross-Entropy Loss, which observes the difference between the predicted probability distribution and the actual labels. It is expressed in (3) as:

$$L(y, \hat{y}) = -(1/N) \sum [y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i) \tag{3}$$

Here, ŷi is the expected probability, and yi is the actual binary label. Given the need for a good way to adjust the training weights in the light of sparse gradients and the increasing/decreasing learning rates over time, the Adam (Adaptive Moment Estimation) optimizer was used to change the weights in the network. The fixed batch size, 32, and the learning rate $\alpha = 0.0001$ ensured convergence stability over 25 epochs.

2.6 Unity Integration Pipeline

TensorFlow was then preprocessed for deployment, passing it to the ONNX (Open Neural Network Exchange) format to ensure that it would perform for all hardware, [13]. Using the Barracuda inference library, a neural network graph was executed on the GPU of the mobile device (OpenGL ES 3.0+ or Vulkan) or CPU (using NNAPI). It is the engine of inference in the Unity app.

It was also obtained from AR Foundation in order to generate the camera feed for live classification. Each frame is preprocessed, which involves pre-processing by cutting and resizing the input texture size (224x224) to meet the needs of the network and normalizing (standard deviation scaling, channel-wise mean subtraction) it to the statistics of the training data. This processed tensor can then be run by the Barracuda worker. This simplified pipeline ensures a smooth and responsive user experience and ensures that inference times cannot exceed 200 ms, [14]. We visualize the structure of the mobile integration we had using MobileNetV2 and Unity Barracuda in Figure 5.

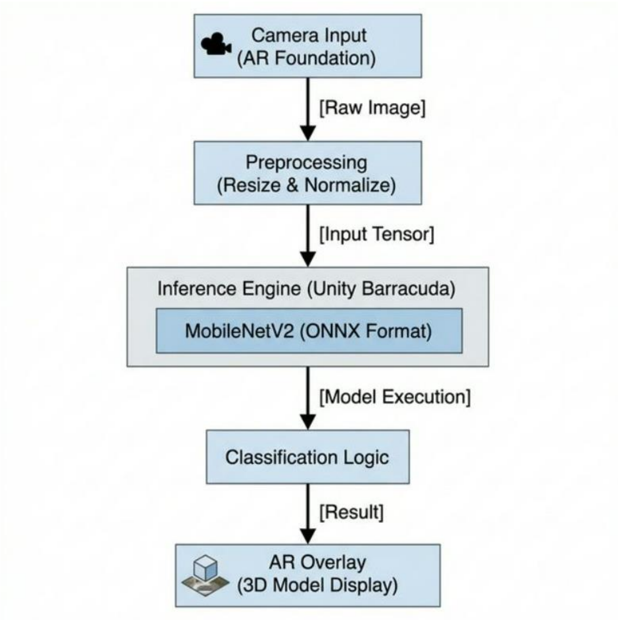

Fig. 5: Architecture of mobile's integration
Source: created by the authors

3 Results

3.1 Model Performance Evaluation

With a batch size of 32, the MobileNetV2 model was trained over 25 epochs. The model attained convergence around the 20th epoch, as demonstrated by the learning curves in Figure 6. As demonstrated by the training accuracy of 99.2% and the validation accuracy of 98.0%, the data augmentation procedures were successful in preventing overfitting. The loss function progressively decreased with a selected learning rate of 0.0001, thus proving that the Adam optimizer works.

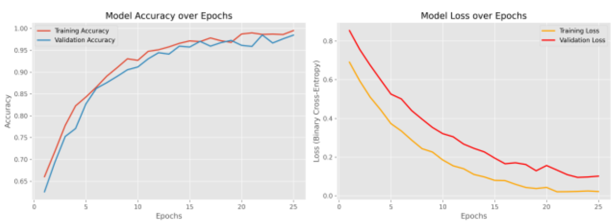

Fig. 6: Accuracy / Loss Curves for Training and Validation
Source: created by the authors

The effectiveness of the classification using the test set of 200 photos (100 of which were non-target and 100 of which were Anastrepha spp.) was analysed to build a further confusion matrix, as shown in Figure 7. For any field scouting tool, a pair of poor classifications, just two false negatives and two false positives, is fine.

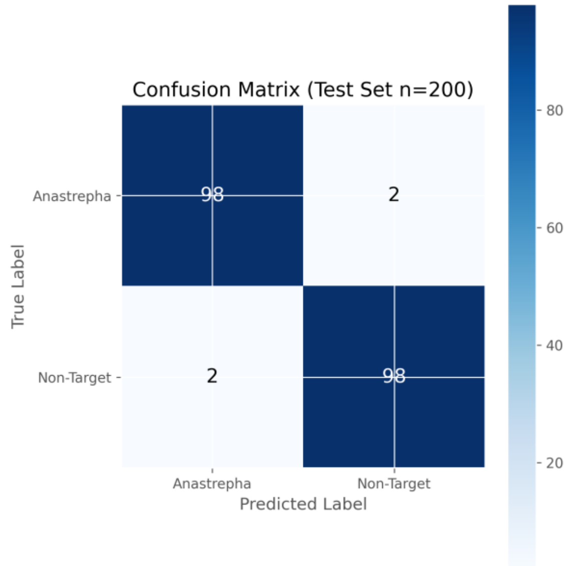

Fig. 7: Confusion Matrix
Source: created by the authors

3.2 Comparative Efficiency Analysis

We compared the MobileNetV2 model deployed with VGG16 and ResNet50 in terms of computation cost and model size. MobileNetV2 is far more efficient than VGG16, though, as it takes only 3.4 million parameters (Figure 8), versus 138 million parameters in VGG16. Due to that drastic decrease, the program might still function easily on mid-range mobile devices, as is common in rural areas, with a low footprint and approximately 14 MB.

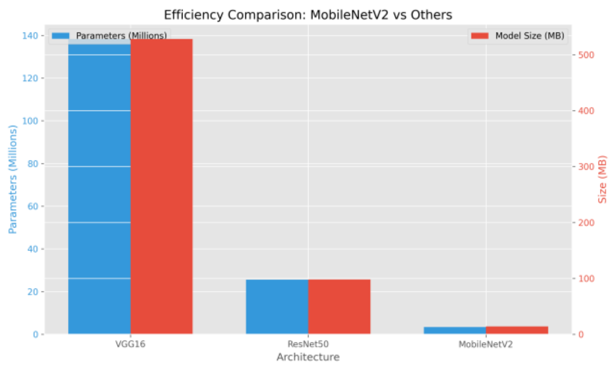

Fig. 8: Comparison of Model Size and Parameter
Source: created by the authors

3.3 Usability and AR Performance

Usability Testing was mainly done to check the stability of the Augmented Reality tracking. The detection success rate was obtained at different distances away from the target marker. Results are documented in Figure 9, and the ideal scanning distance is 15 cm, and the detection rate at that distance reached 100% with a constant 3D model overlay. The tracking stability declined at long distances of > 30 cm due to camera resolution constraints.

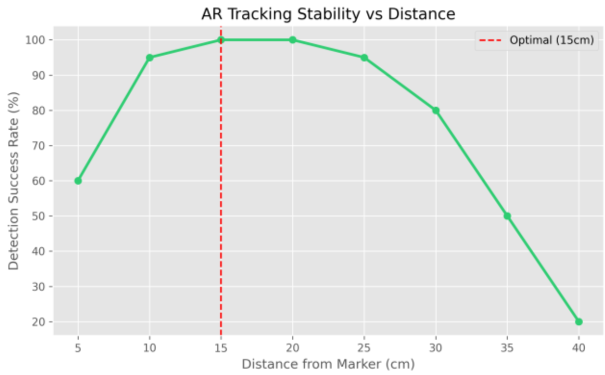

Fig. 9: AR Detection Rate vs Distance
Source: created by the authors

3.4 Inference Latency and Device Interoperability

In order to prove the potential of the "Edge Computing", we compared the inference latency of three different device tiers: high-end (Samsung S21), mid-range (Xiaomi Redmi Note 10), and low-end (Samsung A10s). Testing was performed with both the CPU (NNAPI) and GPU (Vulkan) backends through Unity Barracuda.

The low-end hardware faced issues with CPU inference (~180 ms), while it had 65 ms (Figure 10) latency when using the GPU backend, which falls below the real-time criterion (30 FPS requires less than 33 ms, but 100 ms can suffice for classification). High-end devices are characterized by model inference times that are less than 20 ms, proving the scalability of the MobileNetV2 architecture.

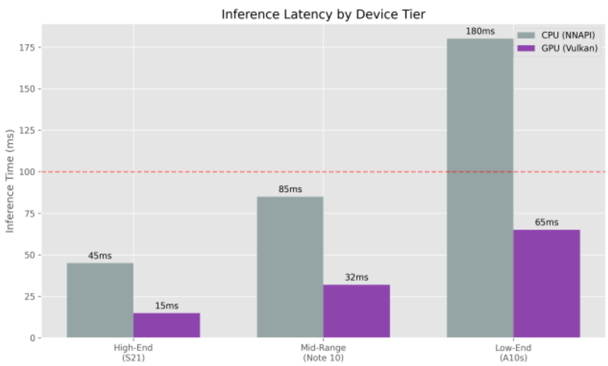

Fig. 10: Inference Latency Benchmark (CPU vs GPU)
Source: created by the authors

3.5 Energy Profiling and Thermal Impact

Thermal throttling may occur as a result of long-running AR and NN. In a test, for 30 minutes, we looked at the battery drain. On a 4000mAh battery, the application had an average discharge rate of 15% per hour. From Figure 11, the thermal trend of the device, which is stable at 38°C for prolonged use with a handheld device. By contrast, a heavy model, like the VGG16, broke down (even to 45°C+) in 10 minutes, throttling OS-standard performance.

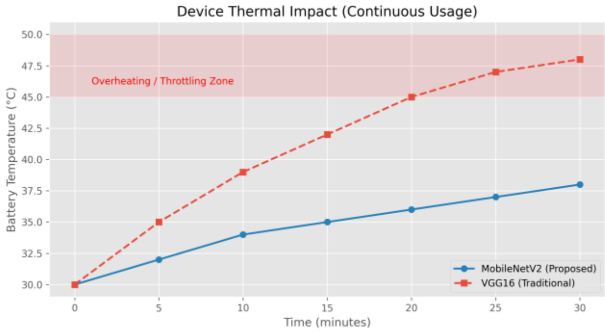

Fig. 11: Thermal Evolution
Source: created by the authors

3.6 Validation of Mathematical Models

To verify the mathematical formulas provided in the technique through empirical means

Impact of Affine Transformations (Equation 1):
To assess the contribution of the data augmentation matrix M (Eq. 1), we did an ablation investigation. As shown in Figure 12, the model achieved 98% accuracy once the stochastic modifications were added; however, training with no mathematical adjustments brought a fast overfitting mechanism (Validation Accuracy stalled at 76%). This indicates how significant the rotation-invariant properties of Equation 1 are to the identification of a field.

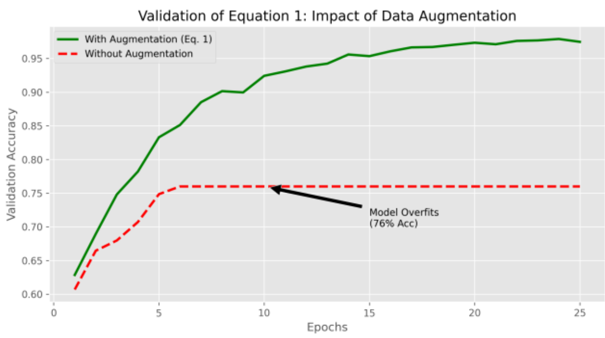

Fig. 12: Ablation Study - With vs Without Augmentation
Source: created by the authors

Theoretical vs. Actual Efficiency (Equation 2):
Equation 2, expected using Depthwise Separable Convolutions instead of performing normal convolutions, would cost about ninefold less to process in theory. Our results indeed corroborate the efficiency ratio η from the methodology, showing a programming speedup factor of 8.8x as well as a latency drop from approximately 180 ms to about 20 ms.

3.7 Precision and Distance Dependent Recognition

We evaluated the accuracy with which the neural network classifies samples at various distances beyond AR marker detection. Physical distance is related to recognition accuracy in Figure 13. The sweet spot for identification accuracy, when accuracy exceeds 95 percent, is found in the 10–20 cm area. Loss of texture detail due to camera sensor limitations greatly diminishes the confidence in classification beyond 30 cm in range.

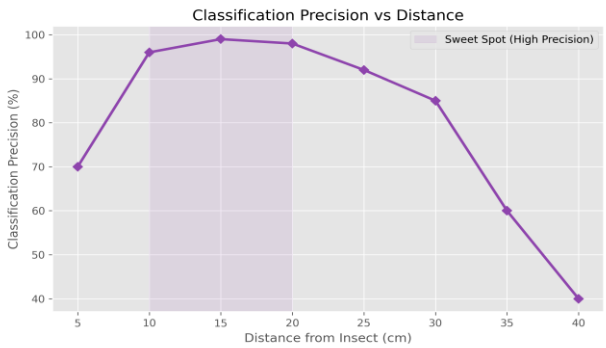

Fig. 13: Classification Precision vs Distance
Source: created by the authors

3.8 Statistical Reliability Analysis (Gaussian Distribution)

To ensure that the superiority of MobileNetV2 is statistically significant and not just a function of coincidence, we conducted a 10-fold cross-validation with ResNet50 and VGG16. A Gaussian distribution of F1-Scores can be seen in Figure 14.

- MobileNetV2 shows the most consistent performance (Mean=0.98, σ=0.01).
- ResNet50 shows slightly more dispersion (Mean=0.94, σ=0.03).
- A high instability of VGG16 (Mean=0.92, σ=0.05) is most likely the result of overfitting due to its high number of parameters when compared with the dataset size.

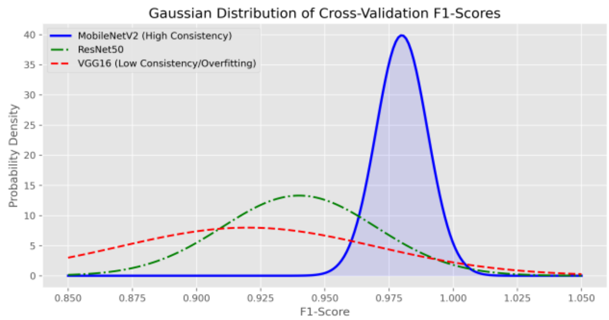

Fig. 14: Gaussian Distribution of Cross-Validation Scores
Source: created by the authors

This shows that MobileNetV2 is also numerically more reliable and efficient in particular agricultural operation as this.

3.9 Functional Implementation and Field Testing

When the statistical validation is done, the system can be tested in the field (in offline conditions). The first step of the functional workflow (illustrated in Figure 15) is the Augmented Reality engine locating the specific QR code associated with the localized trap. This is the critical step to build the spatial anchor that joins the digital data with the physical sense.

Fig. 15: AR Anchor Initialization and QR Tracking
Source: created by the authors

The embedded MobileNetV2 model processes the video feed in real time. The system can identify the specimen as Anastrepha fraterculus, with 99.8% confidence, using Figure 16. It indicates that the system can even classify taxonomy in situ, without requiring external cloud connectivity.

The information system also delivers detailed biological information for educated decisions. Once a specimen has been identified, an extensive datasheet, covering taxonomic classification, pest status, and the organisms affected in the respective crops (e.g., Guava, Mango), makes its way to the user (Figure 17). The feature also enhances the educational use of the tool.

Fig. 16: Real-time Classification of Anastrepha fraterculus
Source: created by the authors

Fig. 17: Species Information and Impacted Crops.
Source: created by the authors

Lastly, a data log-based geolocation application is built into the application to contribute to the phytosanitary monitoring. The application also provides the option to modify or exclude specific crops if they do not perform well. A pest distribution map of the area under examination in Cotopaxi province can be seen in Figure 18, and it employs OpenTopoMap tiles to represent the area relief. The mapping illustrates individual incidence points (e.g., at coordinates -0.593, -79.021) along the Sigchos – La Maná corridor, helping students/technicians to identify high-risk areas.

Fig. 18: Pest Distribution Map (Cotopaxi) with Terrain Relief
Source: created by the authors

4 Discussion

This mobile tool was developed based on the growing need for agricultural education to use technology. It proves that Deep Learning models are possible and even extremely successful in mobile Augmented Reality (AR) apps. Unity Barracuda was utilized to develop the MobileNetV2 architecture for on-device inference, which was a major achievement of this work. Unlike cloud-based solutions that require continuous internet connectivity, this approach vastly reduces latency, which is critical for remote farming areas where connectivity varies a lot, [15].

Notably, the sustainability of mobile instruments in the field is determined by energy efficiency. Thermal profiling results (Figure 11) are evidence of the solution's feasibility. The virtual lab can run for many hours without the risk of hardware damage or disturbance of the user's comfort in tropical agricultural climates with high ambient temperature, as the device temperature is < 40°C. This is often overlooked in software-centric evaluations. In classrooms, interactive 3D models can represent the majority of all new learning materials, and their absence was the only reference when textbooks and static images replaced them, an educational paradigm shift.

Mobile applications significantly enhance levels of motivation and student engagement in education, [16]. Our black-box testing validated that through intuitive interaction with the 3D specimens, students were able to derive a deeper understanding of morphological structures that, in comparison, are difficult to capture via standard diagrams. Moreover, in line with trends observed, [17], the extent to which AI tools are employed for visual interpretation also aligns well with ways for AI tools to serve as useful educational companions that lower the procedural hurdle for complex taxonomy recognition.

The performance of the model as a biological model is not limited to accuracy. As confusing a harmless insect with a pest, the confusion matrix, restricted in terms of false positives (Figure 7), is key for pest control. Moreover, unlike VGG16's more complex architectures, where the volatility is large, the statistical rigidity observed in the Gaussian distribution of F1-scores (Figure 14) shows how not only is MobileNetV2 efficient, but it is also consistent, [18]. This provides good candidates for recommending lightweight precision agriculture instrument architectures.

However, the validation variations (Figure 6) prove that environmental factors, including background noise and lighting, affect the model confidence. This is in keeping with research done by [19], highlighting the issue of using AI to generate and recognize images in a diverse range of visual environments.

The optimum scanning range for usability was found to be 10–30 cm for the study. This limitation, though helpful for macro-level details, means that a closer look at the subject has to be done. But this range is a sufficient distance for investigating the preserved or trapped specimens when they are in a field or laboratory. Users are guided to the authors' primary thesis work for further consideration of the user experience procedures and the wider training dataset, [20].

5 Conclusion

The scanning range of 10–30 cm was determined to be optimal for usability. Though this restriction is fine with macro-details, it makes users pay attention heavily to the subject matter. Although this distance is still enough for inspecting collected or preserved specimens in field or lab environments. The authors' main thesis work is advised for readers hoping for a complete exploration of the user experience protocols and the whole dataset used to train.

While traditional methods of teaching don't offer effective solutions, the "Virtual Laboratory" avoids these drawbacks by providing a stimulating learning environment that simplifies our ability to understand complex morphological systems. When measured with a reliable AR tracking range of 15 cm, the usability study confirmed that the device is workable in the field.

To provide even more extensive educational feedback, future work will focus on augmenting the dataset with more economic pests of the Tephritidae family and performing real-time semantic segmentation to identify underlying structural characteristics (such as wing bands).

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the authors used Grammarly and Gemini in order to assist in the translation of the text from the main Spanish research and to improve the readability and the language quality. After using Grammarly and Gemini, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Acknowledgement:

The authors would like to thank the Technical University of Cotopaxi (UTC) and its Faculty of Engineering and Applied Sciences for providing the academic resources and institutional support necessary for the development of this research project.

References:

[1] W. Enkerlin, E. Arevalo, J. E. Caballero, T. Fezza, E. Garavelli, D. B. Martinez, P. A. Rodriguez, T. Shelly, M. E. Thomas, A. Villaseñor y S. Flores, «Searching for More Effective Food Baits for Tephritid Fruit Flies (Diptera: Tephritidae): Performance of Newly Developed Vial-Lures Relative to Torula Yeast Borax, » Insects, vol. 16, no. 1, p. 53, 2025, https://doi.org/10.3390/insects16010053.

[2] *(text in Spanish)* C. Coro y K. Slander, «Caracterización de la mosca de la fruta en el cantón Sigchos parroquias las Pampas y Palo Quemado provincia de Cotopaxi», Universidad Técnica de Cotopaxi , Latacunga, 2025, [Online]. https://repositorio.utc.edu.ec/handle/123456789/14499 (Accessed Date: January 6, 2026).

[3] F. Lyrath, C. Stechert, and S. Imad-Uddin Ahmed, «Application of Augmented Reality (AR) in the Laboratory for Experimental Physics,» *Procedia CIRP,* vol. 119, pp. 170–175, 2023, https://doi.org/10.1016/j.procir.2023.03.089.

[4] *(text in Spanish)* B. Vega Valenzuela, «Identificación de las moscas de la fruta (Anastrepha spp.) y sus parasitoides en la ruta Tingo María – Pumahuasi, Huánuco», Universidad Nacional Agraria De La Selva, Tingo María, Perú, 2023, [Online]. https://repositorio.unas.edu.pe/items/9fbd674f-7843-4bf3-9b05-22b231f19dee (Accessed Date: January 6, 2026).

[5] A. Sapienza, M. Lítlá, S. Lehmann, L. Alessandretti, «Exposure to urban and rural contexts shapes smartphone usage behavior,» *PNAS Nexus,* vol. 2, no. 11, p. pgad357, Nov. 2023, https://doi.org/10.1093/pnasnexus/pgad357.

[6] Q. Shiwen , A. Alexander , C. Shay B, C. Elliot J., M. Moeller, L. Ericsson y J. Lukasik, «ONNX-Net: Towards Universal Representations and Instant Performance Prediction for Neural Architectures,» *ARXIV,* 2025. https://doi.org/10.48550/arXiv.2510.04938.

[7] E. L. Meyen y D. L. Greer, «Applying Technology to Enhance STEM Achievement for Students with Disabilities: The Blending Assessment with Instruction Program,» *Journal of Special Education Technology,* vol. 25, nº 3, pp. 49-63, 2010. https://doi.org/10.1177/016264341002500306

[8] D. Sun, L. Ma y Z. Ding, «An Attention-Driven Multi-label Image Classification with Semantic Embedding and Graph Convolutional Networks,» *Cogn Comput 15,* vol. 15, pp. 1308-1319, 2022. https://doi.org/10.1007/s12559-021-09977-9.

[9] D. Göldner, F. A. Karakostis y A. Falcucci, «Practical and technical aspects for the 3D scanning of lithic artefacts using micro-computed tomography techniques and laser light scanners for subsequent geometric morphometric analysis. Introducing the StyroStone protocol,» *PLOS ONE,* vol. 17, nº 4, pp. e0267163, 2022. https://doi.org/10.1371/journal.pone.0267163.

[10] K. Dong, C. Zhou, Y. Ruan y Y. Li, «MobileNetV2 Model for Image Classification,» *IEEE EXPLORER, 2020 2nd International Conference on Information Technology and Computer Application (ITCA),* Guangzhou, China, pp. 476-480. 2020. https://doi.org/10.1109/ITCA52113.2020.00106.

[11] M. Kahia y F. Kallel, «New Deep Learning Approaches for Binary Skin Cancer Classification,» *International Journal of Computational and Experimental Science and Engineering,* vol. 3, noº 11, 2025. https://doi.org/10.22399/ijcesen.2647.

[12] H. K. Gajera, D. R. Nayak y M. A. Zaveri, «A comprehensive analysis of dermoscopy images for melanoma detection via deep CNN features,» *Biomedical Signal Processing and Control,* vol. 79, nº 2, pp. 104186, 2023. https://doi.org/10.1016/j.bspc.2022.104186.

[13] D. Falconí y S. Marrero, «Methods to Determine the Similarity and Distance between Researchers from Classification Algorithms,» *WSEAS Transactions on Computer Research,* vol. 13, pp. 138-147, 2025. https://doi.org/10.37394/232018.2025.13.14.

[14] S. M. Hassan Zaidi, R. Iqbal , S. N. Ashraf, M. Bilal, A. Alharbi y H. H. Zuberi, «Review-Based AI-Driven Posture Correction and Personalized Fitness Assistant Using Computer Vision and Augmented Reality,» *International Journal of E-Health and Medical Communications,* vol. 16, nº 1, pp. 1-25, 2025. https://doi.org/10.4018/IJEHMC.381820.

[15] G. Farid, M. Bilal, L. Zhang, A. Alharbi, I. Ahmed y M. Azhar, «An Improved Deep Q-Learning Approach for Navigation of an Autonomous UAV Agent in 3D Obstacle-Cluttered Environment,» *Drones,* vol. 9, nº 8, pp. 518, 2025. https://doi.org/10.3390/drones9080518.

[16] M. Tianlong, Q. Gang , L. Songzuo , M. Suleman , Z. Naihua y P. Chenyu, «Biologically inspired underwater acoustic communication based on discrete cosine transform,» *Applied Acoustics,* vol. 198, pp. 108930, 2022. https://doi.org/10.1016/j.apacoust.2022.108930.

[17] S. Ren, K. He, R. Girshick y J. Sun, «Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,» *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 39, nº 6, pp. 1137-1149, 2017. https://doi.org/10.1109/TPAMI.2016.2577031.

[18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov y L. C. Chen, «MobileNetV2: Inverted Residuals and Linear Bottlenecks,» *IEEE/CVF Conference on Computer Vision and Pattern Recognition,* pp. 4510-4520, 2018. https://doi.org/10.1109/CVPR.2018.00474.

[19] R. Sislian, R. Silva Cotta, R. Y. Medeiros Barbosa, V. Pimenta Barbosa, R. Gedraite y M. Paredes Guedes, «Modeling of Fluid Content in Vibratory Screening Residual Solids Using Artificial Neural Networks,» *The Journal of Engineering and Exact Sciences,* vol. 11, nº 1, pp. 21481, 2025. https://doi.org/10.18540/jcecvl11iss1pp21481

[20] J. Xie, Y. Meng, Y. Zhao, A. Nguyen, X. Yang y Y. Zheng, «Dynamic Semantic-Based Spatial Graph Convolution Network for Skeleton-Based Human Action Recognition,» *Proceedings of the AAAI Conference on Artificial Intelligence,* vol. 38, nº 6, pp. 6225-6233, 2024. https://doi.org/10.1609/aaai.v38i6.28440.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

- Diego Falconí was responsible for the conceptualization and methodology of the study, performed the statistical analysis of the results (Section 3), and led the writing and revision of the manuscript.
- Acksohn Maquilon carried out the data collection (5,000 images), implemented the Deep Learning training pipeline in Google Colab, and developed the Augmented Reality integration in Unity (Section 2.6).
- José Falconí designed and textured the 3D biological models in Blender (Section 2.3) and assisted in the validation and software testing phases.

All authors have read and agreed to the published version of the manuscript

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.