Flower Identifier – Augmented Reality Application for Flower Recognition

Constantin-Sebastian Bolohan
Faculty of Computer Science, “Alexandru Ioan Cuza” University
General Berthelot, No. 16, Iaşi, 700483, Romania
sebastian.bolohan97@gmail.com

Adrian Iftene
Faculty of Computer Science, “Alexandru Ioan Cuza” University
General Berthelot, No. 16, Iaşi, 700483, Romania
adiftene@gmail.com

ABSTRACT
This paper presents a mobile application designed to identify flowers in real time using a combination of deep learning and augmented reality. The system integrates a MobileNetV2 model trained via transfer learning on the Oxford 102 Flowers dataset, enhanced through aggressive image augmentation techniques. The application is built using Unity and AR Foundation for cross-platform support, and uses Firebase for authentication and real-time data management. A chat bot powered by the Gemini API offers botanical explanations in natural language. The app supports live flower recognition through the camera, AR display of relevant information, a personal flower collection, and an interactive assistant. Usability testing revealed a high level of user satisfaction, with an average rating of 4.6/5. The results demonstrate the feasibility of combining AI and AR to create an engaging, educational, and mobile-friendly experience.

Author Keywords
Augmented Reality; Mobile Applications; Flower Recognition; Deep Learning; MobileNetV2; Unity; Firebase

ACM Classification Keywords
I.2.10 [Vision and Scene Understanding]: Modeling and recovery of physical attributes

H.5.1 [Multimedia Information Systems]: Artificial, augmented, and virtual realities

General Terms
Design; Experimentation; Performance; Human Factors

DOI: 10.37789/icusi.2025.9

INTRODUCTION
The rapid development of mobile technologies has enabled the integration of artificial intelligence (AI) and augmented reality (AR) into everyday applications. One practical area where these technologies intersect is plant and flower identification. Traditional mobile applications in this domain often rely on static image processing and predefined datasets, resulting in rigid and sequential user interactions.

This paper presents a real-time mobile application for flower recognition, combining deep learning with augmented reality to offer an immersive and interactive user experience. The system is built using Unity, AR Foundation, and Firebase, and leverages a MobileNetV2 model trained on the Oxford 102 Flowers dataset. Users can scan flowers using their phone camera, view identification results directly in the AR interface, save specimens to a personal collection, and ask questions via a domain-specific chat bot powered by Gemini API.

The goal of this work is to demonstrate that AI and AR can be effectively combined to create a smart, educational, and mobile-friendly tool that enhances informal learning and user engagement.

RELATED WORK
In order to accurately position the proposed solution within the landscape of modern mobile applications, we have analyzed several existing tools that provide plant recognition and diagnostic functionalities. These applications typically rely on convolutional neural networks (CNNs) to process user-captured images and return relevant information such as the plant species, estimated health status, and personalized care recommendations. Among the numerous applications available, we selected three representative examples, **Plantum** [13], **Plantify** [12], and **PictureThis** [11], for a comparative analysis based on features, limitations, and usage models.

Plantum[1] is a mobile application available on both Android and iOS platforms, offering a user-friendly interface and a wide range of features dedicated to plant enthusiasts. Its core functionality consists of identifying plants based on a photograph taken with the phone's camera. Once the scan is complete, the application provides the name of the plant, an estimation of its general health condition, and tailored care recommendations (e.g., optimal light and watering conditions). Additionally, the application keeps a history of previously scanned plants, organized as a personal “collection”. Beyond visual recognition, Plantum includes a diagnostic module for detecting plant diseases, a watering calculator, and the ability to set care reminders. These features are available free of charge; however, the

[1] https://plantum.ro/

application displays advertisements and offers a premium version. The premium subscription removes ads, allows unlimited identifications, and grants access to an extended library of personalized advice and plant care insights.

Plantify[2] is a modern mobile application for plant identification and diagnosis, available for free on both Google Play and the App Store. From the first interaction, the application emphasizes an interactive and educational user experience. Its core functionality, similar to Plantum, allows users to identify plant species based on photos taken with their smartphones. Upon successful identification, the application displays detailed information about the plant, including botanical characteristics, natural habitat, and care recommendations. What sets Plantify apart is the integration of an AI-powered chatbot named AI Botanist, capable of responding in natural language to user queries about plant care, species traits, or general botanical knowledge. This feature enhances user engagement and accessibility, especially for beginners or curious plant lovers. The core features of the app are available at no cost. However, access to unlimited chatbot interactions and advanced plant monitoring tools requires a paid subscription. The premium version also provides users with deeper insights and more accurate diagnostics, positioning the app as a hybrid between an educational assistant and a care companion.

PictureThis[3] is one of the most popular plant identification applications globally, with millions of downloads and a wide user base. Its primary functionality allows users to take a photograph of a plant, after which an AI-based model returns the most likely species, potential diseases, and corresponding treatment suggestions. In addition, the app offers general care tips and allows users to set personalized reminders for watering, fertilizing, and other plant maintenance tasks. A notable feature introduced in recent versions is the ability to discover plant-related articles based on the user's location, using the device's GPS. This function enhances the app's educational value by contextualizing plant information geographically. Moreover, PictureThis has expanded its identification capabilities to include not only flowers and ornamental plants, but also trees, mushrooms, and even birds. While PictureThis offers limited free daily identifications, most of its advanced functionalities—such as detailed disease diagnostics, unlimited scans, and expert consultations—are restricted to its premium tier. As such, the platform positions itself as a professional tool for plant care, albeit with notable limitations in the free version. Although each of the reviewed applications offers a solid set of features—such as plant identification, disease diagnostics, and care reminders—the underlying interaction model remains largely the same: the user opens the app, captures a static photo of the plant, waits for a short period (typically 1–2 seconds), and is then redirected to a results screen containing identification data and care information. While this workflow is functional, it is also passive and sequential in nature. The user is required to perform a manual action (take a photo), followed by a waiting period and a context switch. In contrast, the solution proposed in this paper aims to reduce cognitive friction by enabling live recognition, where plant identification occurs in real time while the camera feed is active. Furthermore, the integration of augmented reality (AR) enhances user engagement by anchoring information directly onto the physical world, offering a more seamless and immersive experience.

This shift from static to dynamic interaction not only shortens response time but also aligns better with the expectations of modern users who value instant feedback, interactivity, and contextual information delivery.

PROPOSED SOLUTION

The proposed system is a mobile application designed for real-time flower recognition, combining deep learning-based image classification with augmented reality and interactive services. This section details the architectural components of the solution, including dataset selection and preprocessing, model development and evaluation, backend and frontend design, and overall system integration.

Technologies Used

TensorFlow[4] was employed for model development due to its scalability and its support for exporting models in the lightweight .tflite format, enabling efficient on-device inference. Alternative frameworks such as PyTorch Mobile or CoreML were considered but were less suitable for seamless Unity integration and cross-platform deployment.

Unity[5], together with AR Foundation, was selected as the development framework because it ensures consistent AR functionality across Android and iOS. Unity Barracuda was integrated to enable .onnx models to run directly on mobile devices, offering a favorable trade-off between accuracy and latency. Other frameworks such as Vuforia or ARCore SDKs provide strong AR capabilities, but lack the same tight integration with Unity's ecosystem and cross-platform UI tools [17].

Firebase[6] was adopted as the backend infrastructure, ensuring authentication, cloud storage, and real-time synchronization. Its ease of integration, scalability, and secure authentication mechanisms made it preferable to alternatives such as AWS Amplify or Supabase. Firebase Cloud Functions were used to securely relay user queries to the Gemini API.

Gemini API[7] was chosen to provide a domain-specific conversational assistant capable of answering botany-related questions in natural language. Compared to alternative large language models, Gemini offered a balance

[2] https://plantify.co.uk/
[3] https://myplantum.com/

[4] https://www.tensorflow.org/
[5] https://unity.com/
[6] https://firebase.google.com/
[7] https://ai.google.dev/

between fluency, contextual accuracy, and ease of integration via REST endpoints, making it suitable for mobile deployment in educational contexts.

Dataset Selection and Preprocessing

In the early stages of the project, two main strategies were evaluated for obtaining a relevant dataset for flower classification tasks: (a) constructing a custom dataset or (b) relying on existing, publicly available datasets.

The first strategy would have involved manually collecting a large number of flower images under varied environmental settings (e.g., natural background, different lighting conditions, diverse angles), followed by human labeling, curation, and preprocessing to ensure uniform resolution and structure. While this method grants complete control over labeling quality, class distribution, and intra- class variation, it poses significant challenges in terms of scalability, data imbalance, and resource demands. Additionally, the limited number of distinct species that could be realistically collected would have resulted in poor generalization and limited model performance.

To mitigate these issues, multiple publicly available datasets were reviewed, including:

- **Flowers Recognition** (Kaggle) – containing 4,000 images across 5 general categories (e.g., daisy, tulip), with moderate resolution but limited taxonomic diversity [6].
- **Flowers Species Dataset** (Kaggle) – featuring high- resolution images in 10 flower categories, offering visual richness but lacking granularity for fine-grained classification tasks [7].

After careful evaluation, the Oxford 102 Flowers [18] dataset was selected as the most appropriate choice. This dataset is widely adopted in academic research and serves as a benchmarking standard for fine-grained flower classification. It contains 8,189 images spanning 102 distinct flower classes, with expert-level annotations, ensuring semantic accuracy and labeling consistency [10].

The dataset is organized into three distinct folders:

- **train**: contains 1,020 images (10 per class),
- **valid**: also contains 1,020 images (10 per class),
- **test**: includes the remaining 6,149 images.

Although this distribution may appear unusual compared to standard practices, it is the recommended split by the dataset authors and is consistently adopted in widely used libraries such as TensorFlow Datasets and PyTorch implementations. The rationale behind this configuration is to ensure robust evaluation by providing a large and diverse test set. To overcome the limitations of the small training set, extensive data augmentation was applied, as detailed below [14]. Each image is associated with a class label represented as an integer in the range 0–101. A separate *.txt* file provides the mapping between class indices and the actual flower names. This folder structure ensures reproducibility and simplifies integration with standard training pipelines. Moreover, the consistency of 10 images per class in training and validation ensures balanced learning dynamics during early training phases. Despite its strengths, the dataset exhibits two main limitations: strong class imbalance in the test set, where some classes are heavily overrepresented and low sample size in the training split, with only 10 images per class. To compensate for these drawbacks, an extensive image augmentation pipeline was applied. Each image from the training set was used to generate multiple variants through the following random transformations:

- Rotation in the range ±30°,
- Zoom up to 20%,
- Horizontal and vertical flipping,
- Brightness adjustment between 70% and 130%.

This augmentation strategy effectively increased the diversity and size of the training data, improving the model's ability to generalize to unseen samples and real- world conditions. The final preprocessed dataset provided a richer input space, robust against background variation, lighting shifts, and pose changes, enabling better performance in live camera scenarios.

Model Architecture and Evaluation

Three different convolutional neural network (CNN) architectures were developed and compared to identify the most suitable model for mobile deployment: a baseline CNN trained from scratch, MobileNetV2 [16] with transfer learning, and EfficientNetB0.

The baseline CNN was implemented using TensorFlow and consisted of three Conv2D-MaxPooling2D blocks with increasing filter sizes (32-64-128), followed by a Flatten layer, a Dense(256) layer with Dropout(0.5), and a Dense(102) softmax output layer. All hidden layers used ReLU activation. The model was trained for 30 epochs using Adam (learning rate: 1e-3) and categorical crossentropy. Despite reaching 60% training accuracy, the model suffered from overfitting, with validation accuracy stagnating around 40%, making it unsuitable for real-world use. To enhance performance, MobileNetV2 was adopted for transfer learning. The pre-trained backbone (ImageNet weights) was augmented with a custom classification head: GlobalAveragePooling2D, Dense(512, ReLU) with L2 regularization, BatchNormalization, Dropout(0.5), Dense(256, ReLU), Dropout(0.5), and a Dense(102, softmax) output layer. The base layers were initially frozen, then the last 20 layers were unfrozen for fine-tuning with a learning rate of 1e-5. Training was performed with a batch size of 16, using image augmentation, EarlyStopping, and ReduceLROnPlateau. The model achieved over 86% validation accuracy with stable convergence, as seen in Figure 1.

EfficientNetB0, another state-of-the-art architecture, was also tested. It employs compound scaling across width, depth, and resolution for balanced performance. A similar custom head as in MobileNetV2 was added, and fine-tuning was conducted on the last 30 layers. While the model also reached 86% accuracy, its larger size and slower inference time made it less optimal for mobile deployment.

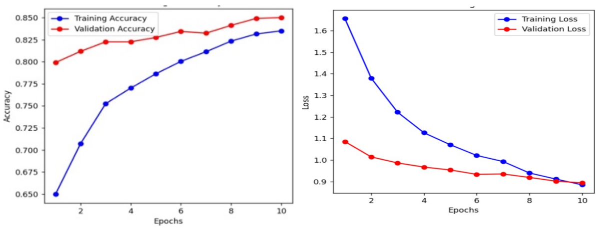

Figure 1. MobileNetV2 accuracy and loss

Consequently, MobileNetV2 was selected for integration into the mobile application due to its favorable trade-off between accuracy, efficiency, and compatibility with Unity's Barracuda inference engine.

Service Architecture

The application integrates several services using the Firebase platform, selected for its high scalability and ease of integration in mobile development environments, as seen in Figure 2 . The core services utilized include:

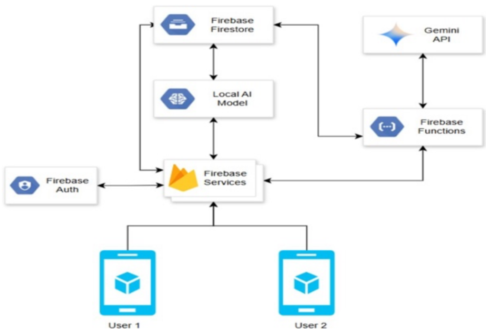

Figure 2. Service architecture diagram of the application

- Firebase Authentication – enables user registration and login. Access to personalized features, such as saving identified flowers, is restricted to authenticated users.
- Firebase Firestore – a real-time NoSQL cloud database used to store and retrieve flower-related information. On-device AI Model – a locally integrated MobileNetV2 model performs real-time flower recognition when the camera is pointed at a plant.
- Gemini API – provides intelligent, AI-powered responses to user queries about flowers and plants via natural language interaction.

Database Structure

Cloud Firestore organizes data into two main collections: "Users" and "Flowers", as seen in Figure 3. Each document in the "Users" collection is keyed by the Firebase UID and contains metadata such as email, display name, timestamps, and login info. Subcollections include: "userFlowers": a list of flower IDs saved by the user and "chatHistory": stores conversational logs with the chatbot. The "Flowers" collection consists of 102 documents, one per class, containing the flower's name, Latin name, habitat, blooming season, description, and an image URL. This schema enables fast data retrieval and user-specific content personalization.

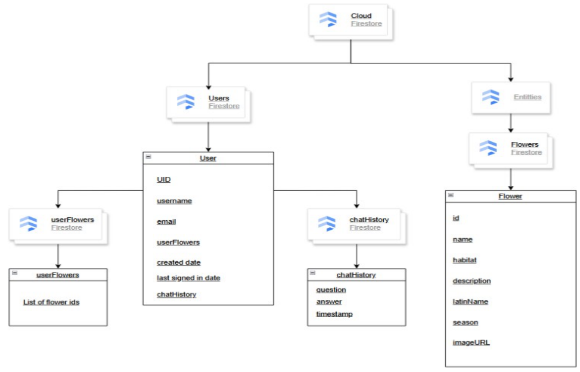

Figure 3. Database Architecture

Application Architecture

The mobile application Flower Identifier was developed in Unity with AR Foundation, combining artificial intelligence, augmented reality, and cloud services. The design emphasizes real-time interaction, scalability, and user-centered experience. Five main modules were implemented: real-time flower recognition, personal collections, chatbot interaction, user profile management, and authentication.

Welcome and Authentication

A standard welcome screen and login flow were implemented to ensure controlled access to personalized features. Unlike generic applications, the design emphasizes both security and usability: Firebase Authentication was used to guarantee encrypted session handling and email verification, while the navigation flow minimizes friction by redirecting authenticated users directly to scanning. These design decisions were motivated by the need to maintain trustworthiness in an educational context, where data integrity and user experience are equally important.

Scan Activity (Augmented Reality)

The scanning module represents the application's central innovation. Unlike related plant-recognition apps, which rely on static photos and separate result screens, our system performs recognition in real time using an on-device MobileNetV2 model. The identified species is displayed together with contextual information—common and Latin name, habitat, season, and description—through world-space AR panels anchored in the camera view. This design fulfills the key criteria of AR systems: combining real and virtual elements, ensuring real-time interaction, and maintaining spatial coherence.

Profile and Personal Collection

The profile module allows authenticated users to build a persistent personal collection of identified flowers, synchronized in real time through Firebase Firestore. This design provides educational continuity: users can revisit their discoveries, which supports informal learning and citizen-science practices.

Chatbot Integration

The chatbot activity extends the recognition feature by enabling natural-language queries. Powered by the Gemini API, it transforms the application into an interactive educational assistant rather than a passive recognition tool. This module was designed to increase user engagement by providing personalized botanical explanations and fostering exploratory learning.

USE CASE SCENARIOS

This section illustrates several real-world use cases in which the proposed application proves useful. Each scenario emphasizes the core functionalities and highlights how different components of the system interact to solve user needs effectively.

Identifying an unknown flower in nature

While taking a walk in the park, a user notices an unfamiliar flower. Curious about its species, the user opens the application and accesses the “Identify Plant” activity, as seen in Figure 4. The camera is activated automatically, and real-time processing begins. Within seconds, the application detects the flower and displays its name, scientific classification, and a brief description. This scenario demonstrates how the combination of on-device inference and an intuitive interface allows users to solve such identification tasks quickly and effortlessly.

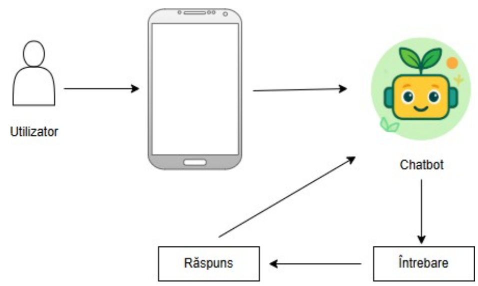

Figure 4. Recognize flower

Building a Personal Collection of Flowers

Over the course of a nature trip, a user identifies multiple flowers using the application. The user decides to save some of them to their personal collection for later reference, as shown in Figure 5. After returning home, they open the “Profile” activity and review the list of saved flowers, each displayed with an image and name. Selecting a flower opens a detail page with comprehensive information. This feature offers users a sense of continuity and allows them to document their discoveries in a structured manner, with data synchronized across devices via the cloud.

Exploring Botanical Curiosities through the Chatbot

Interested in learning more about a particular flower or the differences between two species, the user engages with the chatbot functionality.

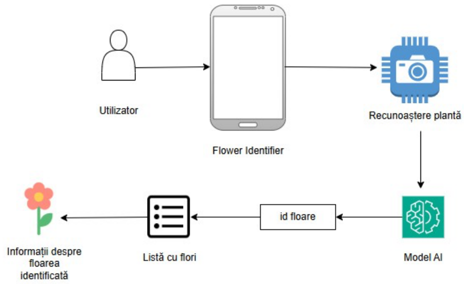

Figure 5. Add identified flower to your collection

By typing a natural language question, the user receives a prompt, well-structured response generated by the Gemini API, as seen in Figure 6. This conversational agent enables exploratory learning and turns the application into an interactive botanical assistant, going beyond simple image classification and offering educational value to curious users.

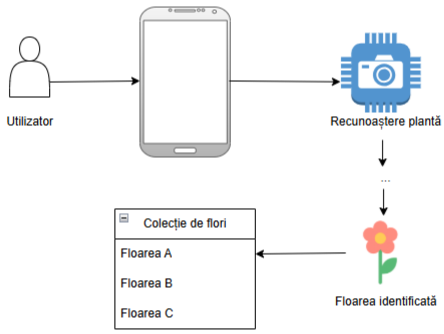

Figure 6. Chatbot communication

USABILITY SECTION

This section presents an overview of user feedback regarding the developed application. In order to evaluate its usability and perceived usefulness, two distinct questionnaires were created: one administered before usage and one after usage. The goal of these surveys was to gather insights on participants’ prior experience with applications involving augmented reality (AR) and visual recognition, as well as to assess their perception of the developed app in terms of interface, clarity, and effectiveness. The questionnaires were distributed to a group of 10 participants, of which 6 were female and 4 were male, aged between 15 and 50, coming from diverse educational and professional backgrounds.

Pre-Use Questionnaire

The initial questionnaire was designed to evaluate participants’ prior exposure to technologies relevant to the application—specifically, augmented reality (AR) and image recognition—as well as to understand their expectations regarding usability and perceived value. This preliminary feedback was essential for establishing a baseline before interaction with the system. The collected data revealed that 70% of respondents were already familiar with AR-based technologies and had previously interacted with similar mobile applications. The most frequently mentioned examples were AR filters available on Instagram and Snapchat, which are widely adopted among casual users. This indicated that, for a significant portion of participants, the concept of interacting with AR-enhanced interfaces was not novel and therefore unlikely to pose cognitive or usability barriers.

Furthermore, the majority of respondents expressed a general comfort and confidence in navigating AR applications, suggesting a high level of technological acceptance. However, some recurring concerns from their past experiences were also identified. These included:

Inaccurate recognition results, especially in cases involving partial occlusion or poor lighting, performance delays, leading to reduced responsiveness or lag in real-time feedback, unclear guidance, especially regarding how to position the camera or interpret on-screen feedback. Such concerns were highly relevant, as they directly informed the design priorities of the proposed system—specifically the need for robust recognition, low-latency inference, and intuitive UI/UX design. *Additionally*, the diversity of technological experience across participants, who ranged in age and background, offered a well-rounded perspective. Some participants had no prior exposure to AR beyond social media filters, while others reported using more advanced tools, such as educational or medical AR applications. This heterogeneity proved to be an advantage in gathering a broad spectrum of usability expectations and in identifying both strengths and areas of potential improvement. Overall, the pre-use questionnaire provided actionable insights that were considered in the application's final design and implementation, contributing to a user-centered development process aligned with contemporary interaction paradigms.

Post-Use Questionnaire

Following hands-on interaction with the application, participants completed a second questionnaire aimed at evaluating the system's usability, design clarity, and overall user satisfaction, particularly in relation to the core functionality of real-time flower recognition. This stage of feedback collection was critical for assessing the perceived effectiveness of the application's user interface and intelligent components.

The results demonstrated a consistently positive reception. Specifically, 60% of respondents awarded the maximum rating of 5 out of 5 for intuitiveness and ease of use, indicating that the application's layout, navigation, and functionality were immediately understandable and user-friendly. The remaining 40% provided a score of 4, suggesting that while their experience was highly satisfactory, there were still minor aspects that could benefit from refinement. Importantly, no participant rated the interface below a 4, underscoring a strong consensus around usability.

When evaluating the flower recognition module, all users reported a positive experience, with particular appreciation for the real-time feedback and seamless integration of AI-based prediction within the augmented reality interface. A total of 60% of participants assigned the maximum score of 5/5, citing quick identification, informative overlays, and the clarity of displayed data (e.g., plant name, habitat, blooming season) as major strengths.

While no critical usability issues were encountered during navigation, several participants offered constructive feedback. Suggestions included:

- *Visual design enhancements*, such as improving button styling and spacing,
- *Further optimization of recognition accuracy*, especially under suboptimal lighting conditions or when the flower was partially obscured,
- *Additional guidance for first-time users* to clarify scanning instructions.

Despite these minor observations, the overall user satisfaction rating was high, with the application receiving a mean score of 4.6 out of 5 across all evaluated criteria. This outcome suggests that the system was perceived as both functional and engaging, validating the design choices made during development. In conclusion, the post-use evaluation confirms that the application effectively meets the expectations of diverse users, offering a robust and enjoyable interface for real-time flower identification and botanical exploration. The insights gathered from this feedback phase are valuable not only for iterative refinement but also for informing future extensions, such as multilingual support, more detailed plant data, and advanced interaction modes.

Post-use questionnaire conclusions

The usability questionnaires successfully captured users' impressions regarding the application's interface, practicality, and the overall experience. Although the testing group was relatively small, it was well-balanced in terms of gender, age, background, and technological exposure, offering useful perspectives across different user profiles.

In summary, the application was perceived as accessible, interesting, and useful. The constructive feedback obtained highlights possible directions for improvement and opens the path toward a more refined and engaging future version.

However, a limitation of this evaluation is the small sample size (n=10), which prevents generalization of the results. Larger-scale usability studies are planned as future work to validate these findings across broader and more diverse populations.

CONCLUSIONS

This work represents a comprehensive blend of both theoretical study and practical implementation in the domain of mobile technologies, with a particular emphasis on the integration of artificial intelligence (AI) and augmented reality (AR). The developed solution goes beyond the creation of a standard mobile application; it embodies a broader vision of how emerging technologies can be leveraged to facilitate informal learning and enrich everyday user experiences. The mobile application presented herein is characterized by its scalability, accessibility, and interactive design. By combining computer vision with real-time AR overlays, the system provides users with instant, context-aware botanical insights. The core functionalities, including real-time flower recognition, the ability to curate a personal collection of identified flowers, and a natural language chatbot interface for botanical queries, collectively offer a multifaceted user experience that merges education, utility, and engagement.

From a technical standpoint, the project demonstrates the effective orchestration of several modern frameworks and tools. The use of Unity ensures cross-platform compatibility and intuitive user interface design. AR Foundation enables robust integration of augmented content within real-world environments. The incorporation of TensorFlow Lite allows efficient on-device inference, crucial for maintaining performance on resource-constrained devices. Furthermore, Firebase services—such as Authentication and Firestore—provide scalable backend infrastructure for user management, data storage, and real-time synchronization.

A key strength of the system lies in its modular architecture, which facilitates future extension and adaptation. Throughout development, special attention was paid to usability, model performance, and the responsiveness of interactions, making the application well-suited for deployment in real-world settings such as educational field trips, botanical gardens, or casual outdoor exploration. Looking ahead, several enhancements are planned to improve the solution's capabilities and expand its reach:

- **Dataset enrichment**: Extending the dataset to shrubs, trees, and medicinal plants would make the application valuable for biodiversity education and citizen-science initiatives.
- **Model refinement**: Employing ensemble models could improve robustness under occlusion or poor lighting.
- **Context-aware features**: Integrating geolocation would allow adaptive information based on regional flora.
- **Localization**: Providing multilingual support would extend accessibility and inclusivity to a global audience.

In conclusion, this work demonstrates the strong potential of combining AI and AR technologies to build intelligent mobile systems that are not only functional, but also educational and immersive. By bridging the gap between technical innovation and user-centric design, the application lays the groundwork for future developments in smart environmental education and plant exploration, with meaningful real-world impact.

REFERENCES

1. DeepMind, "Introducing Gemini," [Online]. Available: https://deepmind.google/technologies/gemini/. [Accessed: July 2025].

2. GeeksforGeeks, "Introduction to Convolution Neural Network," [Online]. Available: https://www.geeksforgeeks.org/introduction-convolution-neural-network/. [Accessed: July 2025].

3. Google, "Firebase Documentation," [Online]. Available: https://firebase.google.com/docs . [Accessed: July 2025].

4. Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, MIT Press, 2016.

5. Goldsborough, P., "A Tour of TensorFlow," arXiv:1610.01178, 2016. [Online]. Available: https://arxiv.org/abs/1610.01178.

6. Kaggle, "Flowers Recognition Dataset," [Online]. Available: https://www.kaggle.com/alxmamaev/flowers-recognition. [Accessed: July 2025].

7. Kaggle, "Flower Species Dataset," [Online]. Available: https://www.kaggle.com/datasets/olgabelitskaya/flower-color-images. [Accessed: July 2025].

8. LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., *"Gradient-based learning applied to document recognition,"* Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

9. Mitchell, T. M., Machine Learning, McGraw-Hill, 1997.

10. Nilsback, M.-E. and Zisserman, A., *"Automated Flower Classification over a Large Number of Classes,"* Proc. Indian Conf. on Computer Vision, Graphics and Image Processing, 2008.

11. PictureThis (Glority Global), "PictureThis – Plant Identifier," [Online]. Available: https://www.picturethisai.com. [Accessed: July 2025].

12. Plantify, "Plant identification and diagnosis," [Online]. Available: https://plantify.io. [Accessed: July 2025].

13. Plantum, "Plant identification mobile application," [Online]. Available: https://plantum.app. [Accessed: July 2025].

14. PyTorch Forums, "Flowers102 train/val/test split," [Online]. Available: https://discuss.pytorch.org/t/flowers102-train-val-test-is-1020-1020-6149-samples-instead-of-6149-1020-1020/220208. [Accessed: July 2025].

15. Russell, S. and Norvig, P., Artificial Intelligence: A Modern Approach, 3rd ed., Prentice Hall, 2010.

16. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L. C., *"MobileNetV2: Inverted Residuals and Linear Bottlenecks,"* in Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 4510–4520, 2018.

17. Sünger, İ. and Çankaya, S., *"Augmented Reality: Historical Development and Area of Usage,"* Journal of Educational Technology and Online Learning, vol. 2, no. 3, pp. 118–133, 2019.

18. TensorFlow Datasets, "Oxford Flowers 102," [Online]. Available: https://www.tensorflow.org/datasets/catalog/oxford_flowers102. [Accessed: July 2025].