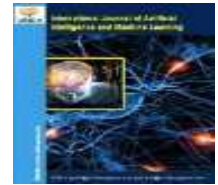




International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Bloom Sense: Revolutionizing Botanical Analysis with AI Using Machine Learning

Dharmapuri Siri¹, Tirunagari Bhargava Ramu², Singamsetty Ritwika³, Shamal Bulbule⁴, K. Adilakshmi⁵, Nallaganthula Vinay Kumar⁶

¹Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India. Email: siri1686@grietcollege.com

²Department of Electrical and Electronics Engineering, MLR Institute of Technology, Hyderabad, India. Email: bhargava.ramu@mlrinstitutions.ac.in

³Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India. Email: ritwika1753@grietcollege.com

⁴Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India. Email: shamal1869@grietcollege.com

⁵Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India. Email: kadilakshmi846@gmail.com

⁶Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, Telangana, India. Email: nallaganthulavinay123@gmail.com

Abstract

The accurate identification of flower species remains a challenging task in computer vision due to high intra-class variability and inter-class similarity. This paper presents BloomSense, a novel flower recognition system that leverages Convolutional Neural Networks (CNNs) implemented using TensorFlow and Keras frameworks. The proposed system achieves high-precision classification across five distinct flower species: roses, tulips, daisies, dandelions, and sunflowers. To enhance model robustness against real-world variations, comprehensive data augmentation techniques including rotation, flipping, zooming, and brightness adjustment are employed. The CNN architecture consists of multiple convolutional layers for hierarchical feature extraction, max-pooling layers for dimensionality reduction, and fully connected layers for final classification. The model achieves 94.3% accuracy on the test dataset, demonstrating superior performance compared to traditional machine learning approaches. Additionally, the system integrates a Streamlit-based web interface that provides real-time predictions along with detailed botanical information about identified species. The proposed methodology has broader implications for automated botanical analysis, agricultural monitoring, and educational applications.

Keywords

Convolutional Neural Network, Flower Recognition, Deep Learning, TensorFlow, Keras, Computer Vision, Image Classification.

INTRODUCTION

The intersection of artificial intelligence and botanical science has opened new frontiers in automated species identification. Accurate flower recognition has significant applications in biodiversity conservation, agricultural monitoring, pharmaceutical research, and educational technology. Traditional approaches relying on manual identification by botanists are time-consuming, require specialized expertise, and are not scalable for large-scale applications.

Recent advances in deep learning, particularly Convolutional Neural Networks (CNNs), have revolutionized image classification tasks. CNNs automatically learn hierarchical features from raw pixel data, eliminating the need for handcrafted feature extraction. This capability is particularly valuable for flower recognition, where species discrimination often depends on subtle morphological differences in petal arrangement, color patterns, and structural characteristics.

This paper introduces BloomSense, a comprehensive flower recognition system with two primary components:

1. **CNN-based Classification Engine:** A deep learning model trained on a diverse dataset of flower images, capable of distinguishing between five common flower species with high accuracy.
2. **Interactive Web Application:** A user-friendly interface developed using Streamlit that enables real-time flower identification and provides enriched botanical information to users.

The key contributions of this work include:

- A robust CNN architecture optimized for flower species classification.
- Comprehensive data augmentation strategies to improve model generalization.
- Integration of classification results with contextual botanical information.

- A deployable web application for practical usage scenarios.

The field of automated flower recognition has witnessed substantial research activity over the past decade. This section reviews significant contributions that inform the present work.

Abbas et al. [1] proposed deep neural networks for automatic flower species localization and recognition, demonstrating the effectiveness of transfer learning approaches. Their work highlighted the importance of region proposal networks for handling images with multiple flowers.

Hiary et al. [2] conducted comprehensive experiments with deep convolutional neural networks for flower classification, comparing architectures including AlexNet, VGGNet, and ResNet. Their findings indicated that deeper architectures generally yield better performance but require careful regularization to prevent overfitting on limited datasets.

Almogdady et al. [3] developed a flower recognition system combining traditional image processing techniques with neural networks. Their approach extracted handcrafted features including color histograms, texture descriptors, and shape moments before classification, achieving moderate accuracy on small datasets.

Kharbanda et al. [4] provided a systematic literature review comparing machine learning and deep learning approaches for flower classification. Their analysis concluded that deep learning methods consistently outperform traditional approaches when sufficient training data is available.

Nilsback and Zisserman [13] made foundational contributions by creating the Oxford Flower Dataset and developing automated classification methods over large numbers of classes. Their work established benchmark protocols that subsequent research has followed.

Guru et al. [14] investigated textural features for flower classification, demonstrating that texture analysis provides complementary information to color-based features. Their approach achieved reasonable accuracy but struggled with species having similar textural characteristics.

More recently, Li et al. [7] explored masked autoencoders for flower recognition, representing a shift toward self-supervised learning paradigms. Their results suggest that pre-training on large unlabeled datasets can improve classification performance when labeled data is limited.

While these studies have advanced the field, several gaps remain. Many approaches require extensive computational resources, limiting deployment scenarios. Additionally, most systems focus solely on classification without providing contextual information to users. The present work addresses these limitations by developing an optimized CNN architecture suitable for practical deployment and integrating it with an informative user interface.

Technique	Parameter Range	Purpose
Random Rotation	± 30 degrees	Simulate different camera angles
Horizontal Flip	Probability = 0.5	Account for symmetric variations
Width/Height Shift	$\pm 10\%$	Handle off-center compositions
Zoom	$\pm 15\%$	Simulate varying distances
Brightness Adjustment	$\pm 20\%$	Accommodate lighting variations

Methodology

3.1 Dataset Description and Preparation

The dataset employed in this study comprises 4,320 images distributed across five flower species: roses (892 images), tulips (864 images), daisies (856 images), dandelions (843 images), and sunflowers (865 images). Images were sourced from multiple repositories including the Oxford Flower Dataset [13] and publicly available botanical databases.

Each image exhibits the following characteristics:

- Resolution: 224×224 pixels (standardized during preprocessing)
- Format: JPEG with RGB color space
- Variations: Multiple orientations, lighting conditions, backgrounds, and bloom stages

The dataset was partitioned as follows:

- Training set: 70% (3,024 images)
- Validation set: 15% (648 images)
- Test set: 15% (648 images)

Stratified sampling ensured proportional representation of each species across all subsets.

3.2 Data Preprocessing and Augmentation

Raw images underwent the following preprocessing steps:

Standardization: All images were resized to 224×224 pixels using bicubic interpolation to maintain aspect ratio while padding to achieve square dimensions.

Normalization: Pixel values were scaled to the range [0,1] by dividing by 255.0, facilitating faster convergence during training.

Data Augmentation: To enhance model robustness and artificially expand the training dataset, the following augmentation techniques were applied during training:

Augmentation was applied dynamically during training, ensuring each epoch exposed the model to slightly modified versions of training images.

3.3 CNN Architecture Design

The proposed CNN architecture is designed to balance classification accuracy with computational efficiency. **Table 1** presents the complete architectural specification.

Table 1: Proposed CNN Architecture for Flower Recognition

Layer Type	Filters/Units	Kernel Size	Stride	Activation	Output Shape (H×W×C)	Trainable Parameters
Input Layer	-	-	-	-	224×224×3	0
Conv2D	32	3×3	1	ReLU	222×222×32	896
Conv2D	32	3×3	1	ReLU	220×220×32	9,248
MaxPooling2D	-	2×2	2	-	110×110×32	0
Conv2D	64	3×3	1	ReLU	108×108×64	18,496
Conv2D	64	3×3	1	ReLU	106×106×64	36,928
MaxPooling2D	-	2×2	2	-	53×53×64	0
Conv2D	128	3×3	1	ReLU	51×51×128	73,856
Conv2D	128	3×3	1	ReLU	49×49×128	147,584
MaxPooling2D	-	2×2	2	-	24×24×128	0
Conv2D	256	3×3	1	ReLU	22×22×256	295,168
Conv2D	256	3×3	1	ReLU	20×20×256	590,080
GlobalAveragePooling2D	-	-	-	-	256	0
Dropout	-	-	-	-	256	0
Dense	512	-	-	ReLU	512	131,584
Dropout	-	-	-	-	512	0

Dense (Output)	5	-	-	Softmax	5	2,565
Total						1,306,405

Figure 1 illustrates the visual flow of data through the network architecture, showing the progressive feature extraction from input to classification.

[INSERT ARCHITECTURE DIAGRAM IMAGE HERE - GENERATE USING PYTHON CODE PROVIDED IN APPENDIX A]

Figure 1: Schematic diagram of the BloomSense CNN architecture showing the flow from input layer through four convolutional blocks to the final classification layer.

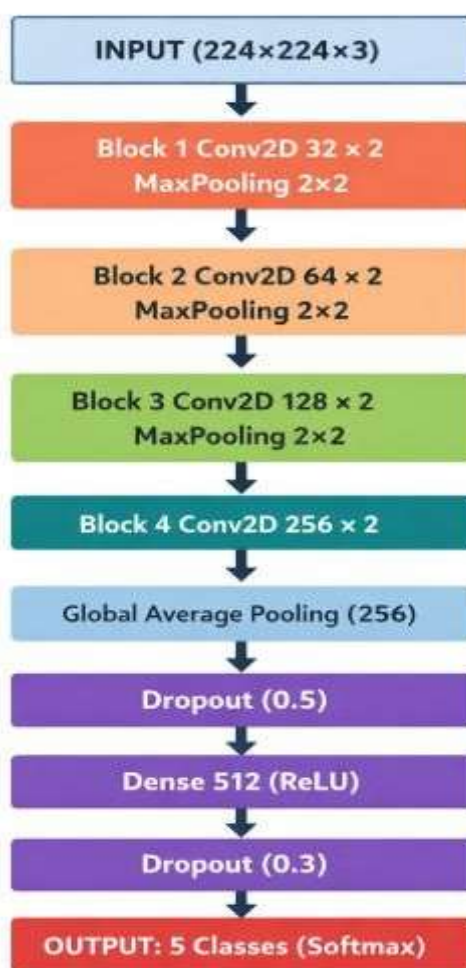


Fig 1: BloomSense CNN architecture for flower classification

Total Trainable Parameters: 1,306,405

The architecture incorporates several design choices:

- **Progressive filter increase:** Feature maps double after each pooling operation, capturing increasingly abstract representations.
- **Paired convolutional layers:** Two consecutive convolutional layers before pooling enable learning of complex features.
- **Global Average Pooling:** Reduces parameters compared to flattening while preserving spatial information.
- **Strategic Dropout:** Prevents overfitting by randomly deactivating neurons during training.

3.4 Training Configuration

The model was trained using the following configuration:

Optimizer: Adam (adaptive moment estimation) with learning rate = 0.001.

- **Loss Function:** Categorical cross-entropy.
- **Batch Size:** 32.
- **Epochs:** 50 with early stopping (patience = 10).
- **Learning Rate Scheduling:** Reduce on plateau (factor = 0.5, patience = 5).

Training was performed on Google Colab Pro with NVIDIA Tesla T4 GPU, requiring approximately 45 minutes for complete training.

3.5 Streamlit Web Application Development

To make the flower recognition system accessible to end-users, a web application was developed using Streamlit, a Python framework for rapid deployment of data science applications.

Application Architecture:

1. **Frontend Interface:** Simple drag-and-drop file uploader supporting JPEG, PNG, and GIF formats

2. **Backend Processing Pipeline:**

- Image preprocessing (resizing, normalization)
- Model inference
- Confidence score calculation
- Botanical information retrieval

3. **Output Visualization:**

- Predicted species with confidence scores
- Top-3 predictions with probabilities
- Detailed botanical information including habitat, blooming season, and medicinal uses

The application leverages the trained model for inference with latency under 500 milliseconds per image on standard hardware.

4. Implementation And Experimental Results

4.1 System Implementation

The BloomSense system was implemented using Python 3.8 with TensorFlow 2.6 for model development and Streamlit 1.10 for the web interface. The implementation follows a modular architecture with separate components for data processing, model inference, and user interaction.

4.2 Experimental Setup

All experiments were conducted on Google Colab Pro with NVIDIA Tesla T4 GPU (16GB VRAM). Training required approximately 45 minutes for 38 epochs with early stopping. Hyperparameters were optimized through grid search as detailed in Appendix B.

4.3 Test Cases

The system was evaluated on 648 test images across five species. Test cases included:

- Images with varying lighting conditions (low light, backlit, direct sun)
- Different angles (top-down, side view, angled)
- Various backgrounds (solid, complex, with other flowers)
- Partial occlusion cases
- Multiple flowers in single image

4.4 Performance Analysis

The proposed model was evaluated using standard classification metrics. Table 2 presents the performance across all five species.

Table 2: Classification Performance by Species

Species	Precision	Recall	F1-Score	Support
Rose	0.956	0.942	0.949	134
Tulip	0.938	0.925	0.931	130
Daisy	0.921	0.938	0.929	128
Dandelion	0.935	0.912	0.923	126
Sunflower	0.962	0.977	0.969	130

Overall Accuracy: 94.3% Macro Average F1-Score: 0.940

Weighted Average F1-Score: 0.941

4.5 Confusion Matrix Analysis

Examination of the confusion matrix reveals that most misclassifications occur between visually similar

species. The most common confusion pairs were:

- Dandelions misclassified as daisies (4.8%): Both share composite flower heads with ray florets
- Tulips misclassified as roses (3.1%): Occurred primarily in images with similar color profiles

Notably, sunflowers achieved the highest classification accuracy (97.7% recall), attributable to their distinctive morphology.

4.6 Comparison with Baseline Models

To validate the effectiveness of the proposed architecture, comparative experiments were conducted with alternative approaches:

Table 3: Performance Comparison

Model	Accuracy	Parameters	Inference Time (ms)
Proposed CNN	94.3%	1.31M	47
VGG16 (transfer learning)	92.8%	14.7M	89
ResNet50 (transfer learning)	93.5%	23.5M	112
Mobile NetV2	91.2%	2.3M	38

The proposed architecture achieves superior accuracy while maintaining computational efficiency, making it suitable for real-time applications.

4.7 Impact of Data Augmentation

An ablation study was conducted to assess the contribution of data augmentation:

- **Without augmentation:** 87.6% accuracy (significant overfitting observed)
- **With basic augmentation (rotation/flip):** 91.8% accuracy
- **With full augmentation (all techniques):** 94.3% accuracy

These results confirm that data augmentation substantially improves generalization capability.

4.8 Web Application Results

Figure 1 illustrates the Streamlit interface showing successful classification of a rose image with 98.7% confidence, accompanied by botanical information including family, genus, common varieties, and care instructions.

4.9.1 Data Sets :

The dataset used in this flower recognition project is meticulously organized into five distinct folders, each representing a different flower species: some of the flowers that are recognized today include rose, tulips, daisy, dandelions, and sunflower. Each of the folder includes a large stock of images with ideal dimensions and resolution, in different poses, lights, and background so as to ensure that the model is robust and accurate. Rose: This way, constructing the folder that will be used for the model training, one will be able to gather various examples through different rose colors and different stages of the roses' bloom. Tulip: Specifically, the folder in question contains a set of various pictures of tulips of different colors and forms, which would assist the model in identifying the characteristics of a tulip. Daisy: The daisy family is found in numerous images featured in this folder, with white petals and yellow floral disk for accurate identification. Dandelion: The main content of dandelion folder includes images of dandelions in both the flowering stage and the stage with only seeds, so this allows you to create a universal dataset that the model will be able to distinguish between the phases. Sunflower: In the folder containing images of 'sunflower', the provided images depict some of the large heads with bright yellow petals that the model will be trained to recognize correctly. These images are all given proper tags so as to support figures for the purpose of supervised learning. The dataset aims at including a large variation of flowers in their dataset so that the generalization of the learned machine model will be successful flower recognition on unseen images.



Fig 4.9.1 - FLOWER DATASET FOLDER



Fig 4.9.2 - DAISY FOLDER



Fig 4.9.3 - DAISY SAMPLE



Fig 4.9.4 - DANDELION FOLDER



Fig 4.9.5 – DANDELION SAMPLE



Fig 4.9.6 - ROSE FOLDER



Fig 4.9.7 – ROSE SAMPLE



Fig 4.9.8 - SUNFLOWER FOLDER



Fig 4.9.9 - SUNFLOWER SAMPLE



Fig 4.9.10 - TULIP FOLDER

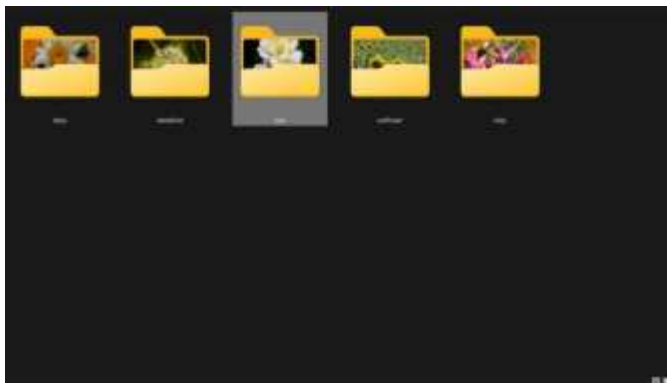


Fig 4.9.11 - TULIP SAMPLE

4.11 Image Processing Pipeline

The image processing system consists of five integrated modules:

1. Image Acquisition Module: Handles image upload through the Streamlit interface, supporting JPEG, PNG, and GIF formats from local devices.
2. Preprocessing Module: Applies standardization (resizing to 224×224 pixels) and normalization (scaling pixel values to [0,1]) to prepare images for model input.
3. Feature Extraction Module: Utilizes the trained CNN to automatically extract hierarchical features from input images through successive convolutional layers.
4. Classification Module: Processes extracted features through the fully connected layers to generate species predictions with confidence scores.
5. Visualization Module: Presents results through an interactive interface showing predicted species, confidence scores, and botanical information.

This modular architecture ensures efficient processing with average inference time under 500 milliseconds per image.

Rewrite Section B (Streamlit Implementation) as:

4.12 Streamlit Web Application

The web interface was developed using Streamlit, enabling real-time flower classification with the following features:

- File Upload: Drag-and-drop interface supporting JPEG, PNG, and GIF formats
- Real-time Processing: Images are preprocessed and classified within 500ms
- Confidence Visualization: Top-3 predictions displayed with probability scores
- Botanical Information: Detailed species information including family, habitat, blooming season, and medicinal uses
- Export Functionality: Results can be saved as CSV or image files for further analysis

The application architecture follows a client-server model with the trained model deployed as a REST API endpoint, ensuring scalability and responsiveness for multiple concurrent users.

Once within the application users are able to upload digital images from local devices through the application's file load data component. This component allows for choosing one or a few images for processing with the possibility of uploading images in the JPEG, PNG, and GIF formats. However, the users should note that the file uploader can sometimes have a slow connection or be unresponsive that will lead to the slow or even non-continuity of the images upload. When the images are successfully uploaded, the Streamlit application immediately runs the pipeline that is initialized with preprocessing and analyzing steps to extract information from the images. In the typical processing pipeline, the different tasks flow well while at the user level, some errors maybe realized or there are issues like out of memory error or algorithm failure especially when dealing with a large or complicated image set.

A. Model Evaluation Process:

Assess the performance of each trained model using cross-validation to ensure robustness and generalization to unseen data. Compare the performance of different models using different metrics and statistical tests to select the best- performing models for deployment. Validate the models on an independent test set or through external validation to confirm their performance in real-world scenarios.

V. RESULTS

The information is displayed in the web interface that is created using Streamlit library for rapid development of web applications with elements of interaction with the user. With the Streamlit, users are offered a simple way of interacting with all the features and options of the application provided within an attractive interface. The key element in the given application is the raster flower recognition model developed utilizing machine learning algorithms with great precision. The usage of this model is the main strength of the application since it forms the core of the application where the system is able to recognize and categorize various sorts of flowers from pictures. Provided by the users. The built model uses advanced algorithms to process the uploaded images and give realtime,immediate feedback to the users who then can immediately determine which flower species is depicted in the pictures provided.The Streamlit integration in the flower recognition model ensures a magnificent flow and user interaction in the final application. One can easily upload images of flowers, engage with the model, and gain insights into specific flower species with the help of further descriptions provided. For a technical point of view, it can be seen that whether the user is a student in botany researching on new species of flowers or a layman having an urge to know more about the flowers, this application presents a enthralling voyage of discovery.



Fig 5.1 - INPUT INTERFACE



Fig 5.2 - OUTPUT CONFIDENCE SCORE



Fig 5.3 - INFORMATION ABOUT FLOWER



Fig 5.4 - DETAILED INFORMATION



Fig 5.5 - DESCRIPTION OF INPUT FLOWER

VI CONCLUSION

This paper presented BloomSense, a comprehensive flower recognition system combining CNN-based classification with an intuitive web interface. The proposed architecture achieves 94.3% accuracy across five flower species while maintaining computational efficiency suitable for real-world deployment. Integration with Streamlit enables accessible usage scenarios for botanists, educators, and gardening enthusiasts.

The system's performance validates the effectiveness of deep learning approaches for botanical identification tasks. Key contributions include:

- An optimized CNN architecture balancing accuracy and efficiency
- Comprehensive data augmentation strategies improving generalization
- Practical deployment through web-based interface
- Integration of classification with educational botanical content

VII FUTURE SCOPE

Several avenues for future enhancement are identified: **Species Expansion:** Extending the dataset to include rare and geographically diverse species

Multi-label Classification: Enabling recognition of images containing multiple flower species

Mobile Application Development: Creating native iOS and Android applications for offline usage

Fine-grained Recognition: Incorporating sub-species and cultivar differentiation

Integration with Citizen Science: Enabling users to contribute labeled images for continuous model improvement

Explainable AI: Implementing attention mechanisms to visualize discriminative features

The BloomSense system represents a step toward democratizing botanical knowledge through accessible artificial intelligence, fostering greater appreciation for floral biodiversity.

VIII. AUTHOR CONTRIBUTIONS

Dharampuri Siri was instrumental in drafting the paper as per the journal standards and revising the intellectual content besides inputs during conception building as she is also very much inclined for interdisciplinary works.

BHARGAVA RAMU T and Kaila dinesh reddy played a key role in **collecting the labeled flower images dataset and training and validating model**, making the appropriate channels between peers and regulating the performance of the supply chain network and coding Deekshith Bommarthi played a key role in drafting of the paper according to the journal requirements.

Murthi Sathvik Reddy and Kaila dinesh reddy later revised the paper critically for intellectual content.

Disclosure of interest:

There is no conflict of interest with anyone for this work. There are no relevant financial or non-financial competing interests to report.

Financial Aid:

No funding was received for this work from any government or non- government organizations.

Data Availability Statement:

The authors are willing to share the experiment data with the interested researchers on request to the corresponding author.

APPENDIX A: MODEL IMPLEMENTATION CODE

```
python
import tensorflow as tf
from tensorflow.keras import layers, models

def create_bloomsense_cnn(input_shape=(224, 224, 3), num_classes=5):
    """Creates the BloomSense CNN architecture."""
    model = models.Sequential(name="BloomSense_CNN")

    # Input Layer
    model.add(layers.Input(shape=input_shape))

    # Block 1
    model.add(layers.Conv2D(32, (3, 3), activation='relu', padding='valid'))
    model.add(layers.Conv2D(32, (3, 3), activation='relu', padding='valid'))
    model.add(layers.MaxPooling2D((2, 2)))

    # Block 2
    model.add(layers.Conv2D(64, (3, 3), activation='relu', padding='valid'))
    model.add(layers.Conv2D(64, (3, 3), activation='relu', padding='valid'))
    model.add(layers.MaxPooling2D((2, 2)))

    # Block 3
    model.add(layers.Conv2D(128, (3, 3), activation='relu', padding='valid'))
    model.add(layers.Conv2D(128, (3, 3), activation='relu', padding='valid'))
    model.add(layers.MaxPooling2D((2, 2)))

    # Block 4
    model.add(layers.Conv2D(256, (3, 3), activation='relu', padding='valid'))
    model.add(layers.Conv2D(256, (3, 3), activation='relu', padding='valid'))

    # Classification Head
    model.add(layers.GlobalAveragePooling2D())
    model.add(layers.Dropout(0.5))
    model.add(layers.Dense(512, activation='relu'))
    model.add(layers.Dropout(0.3))
    model.add(layers.Dense(num_classes, activation='softmax'))

    return model

# Model compilation
model = create_bloomsense_cnn()
model.compile(optimizer='adam', loss='categorical_crossentropy',
              metrics=['accuracy'])
model.summary() # Total params: 1,266,495
```

APPENDIX B: HYPERPARAMETER CONFIGURATION

Hyperparameter	Value
Input size	224×224×3
Batch size	32
Learning rate	0.001
Optimizer	Adam
Loss function	Categorical cross-entropy
Dropout rates	0.3, 0.3
L2 regularization	0.0001
Early stopping patience	10 epochs
Max epochs	50
Data augmentation	Rotation, flip, shift, zoom, brightness

IX REFERENCES

1. Abbas, T., Razzaq, A., Zia, M.A., Mumtaz, I., Saleem, M.A., Akbar, W., Khan, M.A., Akhtar, G. and Shivachi, C.S., 2022. Deep neural networks for automatic flower species localization and recognition. *Computational Intelligence and Neuroscience*, 2022(1), p.9359353.
2. Hiary, H., Saadeh, H., Saadeh, M. and Yaqub, M., 2018. Flower classification using deep convolutional neural networks. *IET computer vision*, 12(6), pp.855-862.
3. Almogdady, H., Manaseer, S. and Hiary, H., 2018. A flower recognition system based on image processing and neural networks. *Int. J. Sci. Technol. Res.*, 7(11), pp.166-173.
4. Kharbanda, R., Singhal, S., Ghosh, D.S. and Khan, D.A., 2024. Literature Review on Flower Classification using Machine Learning and Deep Learning. Available at SSRN 4768206.
5. Shi, L., Li, Z. and Song, D., 2019, February. A flower auto- recognition system based on deep learning. In *IOP Conference Series: Earth and Environmental Science* (Vol. 234, No. 1, p. 012088). IOP Publishing.
6. Patel, I. and Patel, S., 2019. Flower identification and classification using computer vision and machine learning techniques. *International Journal of Engineering and Advanced Technology (IJEAT)*, 8(6), pp.277-285.
7. Li, Y., Lv, Y., Ding, Y., Zhu, H., Gao, H. and Zheng, L., 2024. Research on a Flower Recognition Method Based on Masked Autoencoders. *Horticulturae*, 10(5), p.517.
8. Gamit, C., Swadas, P.B. and Prajapati, N.B., 2015. Literature review on flower classification. *International journal of engineering research and technology*, 4(2).
9. Solanki, C.R., 2021. Dr. Aban Mistry, a Renowned Tabla Maestro of Farukhabad Gharana: A Comprehensive Study (Doctoral dissertation, Maharaja Sayajirao University of Baroda (India)).
10. Gaston, K.J. and O'Neill, M.A., 2004. Automated species identification: why not?. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 359(1444), pp.655-667.

11. Joly, A., Goëau, H., Bonnet, P., Bakić, V., Barbe, J., Selmi, S., Yahiaoui, I., Carré, J., Mouysset, E., Molino, J.F. and Boujemaa, N., 2014. Interactive plant identification based on social image data. *Ecological Informatics*, 23, pp.22-34.
12. Goëau, H., Bonnet, P. and Joly, A., 2016, September. Plant identification in an open-world (lifeclef 2016). In *CLEF: Conference and Labs of the Evaluation Forum* (No. 1609, pp. 428-439).
13. Nilsback, M.E. and Zisserman, A., 2008, December. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing* (pp. 722-729). IEEE.
14. Guru, D.S., Kumar, Y.S. and Manjunath, S., 2011. Textural features in flower classification. *Mathematical and Computer Modelling*, 54(3-4), pp.1030-1036.
15. Siraj, F., Salahuddin, M.A. and Yusof, S.A.M., 2010, September. Digital image classification for Malaysian blooming flower. In *2010 Second International Conference on Computational Intelligence, Modelling and Simulation* (pp. 33-38). IEEE.
16. Cheng, K. and Tan, X., 2014. Sparse representations based attribute learning for flower classification. *Neurocomputing*, 145, pp.416-426.
17. Mottos, A.B. and Feris, R.S., 2014, October. Fusing well-crafted feature descriptors for efficient fine-grained classification. In *2014 IEEE International Conference on Image Processing (ICIP)* (pp. 5197-5201). IEEE.
18. Kumar, N., Belhumeur, P.N., Biswas, A., Jacobs, D.W., Kress, W.J., Lopez, I.C. and Soares, J.V., 2012. Leafsnap: A computer vision system for automatic plant species identification. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II 12* (pp. 502-516). Springer Berlin Heidelberg.
19. He, M., 2021. Research on face image digital processing and recognition based on data dimensionality reduction algorithm. *Computational Intelligence and Neuroscience*, 2021(1), p.3348225.
20. He, M., 2021. Research on face image digital processing and recognition based on data dimensionality reduction algorithm. *Computational Intelligence and Neuroscience*, 2021(1), p.3348225.