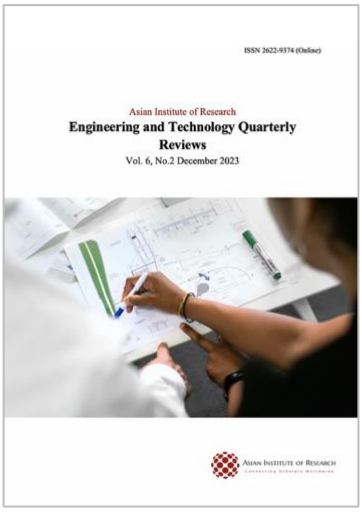

Engineering and Technology Quarterly Reviews

Kelvianto, K., & Kusnadi, A. (2023), Design and Development of an Android-Based Flower Classification Application Using Artificial Neural Networks with Backpropagation Method. In: *Engineering and Technology Quarterly Reviews*, Vol.6, No.2, 33-38.

ISSN 2622-9374

Published by:
The Asian Institute of Research

The Asian Institute of Research
Engineering and Technology Quarterly Reviews
Vol.6, No.2, 2023: 33-38
ISSN 2622-9374

Design and Development of an Android-Based Flower Classification Application Using Artificial Neural Networks with Backpropagation Method

Kevin Kelvianto[1], Adhi Kusnadi[2]

[1,2] Informatics department/Engineering and Informatics faculty, Universitas Multimedia Nusantara, Tangerang, Indonesia. Email: Adhi.kusnadi@umn.ac.id

Abstract
This study investigates the use of artificial neural networks (ANN) employing the backpropagation method to identify flower types based on petal shapes. Android devices were utilized to capture flower images and transmit them directly to a server. Once a sufficient number of images were gathered, the training of the artificial neural network commenced. The flower images were processed to extract shape features using Sobel edge detection followed by thresholding. Subsequently, the data were normalized and fed into the ANN for training. Once the training was complete, the Android devices were capable of capturing new flower images and using the ANN to identify them. The findings of this research indicate that by using a single hidden layer with 35 hidden nodes, the system achieved a flower detection accuracy of 80%.

Keywords: ANN, Backpropagation, Flower Classification

1. Introduction

Flowering plants, also known as angiosperms, are estimated to have over 200,000 different species (Shrestha et al., 2018). Botanists and plant experts can identify these plants based on their flowers, owing to their extensive training and familiarity with floral characteristics. However, for the average individual without specialized knowledge in botany, recognizing and distinguishing between different flower species can be challenging. Most people are familiar with only a few common types of flowers. To identify a specific flower species, one often has to consult a floral encyclopedia or search online using relevant keywords. Such methods can be cumbersome, especially if one frequently encounters unfamiliar flowers.

Today almost 3.6 billion users (around 45% of the world population and 67% of total mobile phone users) among 5.26 (around 67% of the world population) billion unique mobile phone users of 10.4 billion mobile connections all over the world use smartphone devices (Aznan et al., 2017). People carry their smartphones wherever they go, to the extent that these devices have become integral to their daily lives. Given the widespread use of smartphones, they present a convenient platform for flower identification. Imagine if simply taking a photograph of a flower with a smartphone could identify its species; it would significantly simplify and expedite the identification process. Android, as a mobile operating system, dominates the current market share (Jaiswal, 2018).

This study draws inspiration from previous research, notably (Hiary et al., 2018), in this study, the authors propose a novel two-step deep learning classifier to distinguish flowers of a wide range of species. First, the flower region is automatically segmented to allow localisation of the minimum bounding box around it. The proposed flower segmentation approach is modelled as a binary classifier in a fully convolutional network framework. Second, they build a robust convolutional neural network classifier to distinguish the different flower types. (Kuznetsova et al., 2020)(Rao et al., 2021), in this paper, present the Global Filter Network (GFNet), a conceptually simple yet computationally efficient architecture, that learns long-term spatial dependencies in the frequency domain with log-linear complexity. The architecture replaces the self-attention layer in vision transformers with three key operations: a 2D discrete Fourier transform, an element-wise multiplication between frequency-domain features and learnable global filters, and a 2D inverse Fourier transform. We exhibit favorable accuracy/complexity trade-offs of our models on both ImageNet and downstream tasks This current research differentiates itself by leveraging edge detection for feature extraction and utilizing Android-based smartphones for both image capturing and direct processing. In another study conducted by Bowo, the template matching method was employed to differentiate input images from database templates. This process involved feature extraction via edge detection followed by image thinning.

Given these advancements and the potential of smartphones, our research aims to build a user-friendly and efficient flower classification application based on neural networks, focusing on harnessing the Android platform for real-time processing.

2. Related Theory

2.1. Edge Detection

Edge Detection serves as a pivotal technique in image processing, aiming to identify the boundaries within an image (Marmanis et al., 2018). At the core of this approach lies the principle of detecting significant intensity shifts between adjacent regions. An edge can be conceptually defined as "a connected set of pixels that mark the boundary between two distinct regions" (Waldner & Diakogiannis, 2020). This demarcation often holds essential information about the object, encapsulating its shape and size, making edge detection invaluable in various applications.

The core idea behind edge detection revolves around the examination of pixel intensity changes. When an image is processed, areas where pixel intensity changes abruptly indicate potential edges. These changes can be mapped and identified, producing a simplified version of the image that highlights these critical boundaries.
Edge detectors primarily leverage two types of detectors: row detectors (Hy) and column detectors (Hx). Several operators fall into these categories, with prominent examples being the Roberts, Prewitt, Sobel, and Frei-Chen operators (Meester & Baslamisli, 2022). These operators use mathematical functions to compute the gradient magnitude at each point in the input image, thus providing a measurement of the edge strength.

2.2. Artifial Neural Networks

Artificial neural networks (ANN) are computer models that draw inspiration from the brain's neural architecture (Nwadiugwu, 2020)(Kusnadi et al., 2022). They seek to imitate the brain's capacity for pattern recognition and data-driven learning. ANNs' capacity to adapt to and learn from the data they are trained on has led to their use in a wide range of fields, including image identification, natural language processing, financial forecasting, and more.

Basic Elements:

- Neuron: The basic unit of ANN that receives inputs and produces outputs.
- Weights: Values that determine the strength of the connection between neurons.
- Bias: A constant value added to the result of multiplying input with weights.
- Activation Function: A function that transforms the linear combination of weights, inputs, and bias into the neuron's output.

There are several commonly used ANN architectures:

- Feedforward Neural Network: A network where information only moves forward, from input to output.
- Recurrent Neural Network (RNN): A network where information can move backward, allowing memory of previous inputs.
- Convolutional Neural Network (CNN): Specifically for data with spatial structures, like images.

ANN is a powerful tool in the field of machine learning and artificial intelligence. Despite its drawbacks, with proper understanding and application, ANN can be used to solve various complex problems in various fields.

3. Method

The design and development of the proposed artificial neural network (ANN) for recognizing flower types involve distinct stages of training and recognition (Kusnadi et al., 2023). These are elucidated as follows:

1. Training Phase
   The foundation of the training phase rests on the Backpropagation algorithm. The principal objective is to adjust the weights to obtain an optimal neural network configuration. Post-training, the ANN's configuration, including its weights, will be stored in a dedicated file. The weights are adjusted dynamically based on predefined learning rates and momentum rates until the network achieves the desired recognition accuracy for the input data.

2. Recognition Phase
   This phase exclusively focuses on the forward propagation process. The stored ANN configuration file is retrieved, utilizing the weights for data recognition.
3. Image Pre-processing
   Input images undergo initial processing, beginning with edge detection employing the Sobel operator.
   Subsequent resizing accentuates the image clarity through thresholding techniques.
   Every pixel undergoes transformation into a double sequence, ranging between [0,1], derived from the average R, G, B values of each pixel.
4. Data Representation
   The output count of the ANN is automatically determined by the number of flower types present in the database.
   Database indices are converted to binary format, forming the target output sequence.
5. Architectural Details
   The ANN is structured across three layers, as illustrated in Figure 1. The input layer is optimized for images of resolution 150x150 pixels, translating to 22,500 required inputs. A single hidden layer is introduced, the neuron count of which will be ascertained post-identification of the optimal network architecture. The link between the input and hidden layers will be enhanced using the Nguyen and Widrow (1990) optimization method. To expedite the training process, momentum - the additional weight from the subsequent weight change - is employed. Initial base weights will be randomly set within the range [-0.5, 0.5]. Output count will be automatically generated during training based on available data.

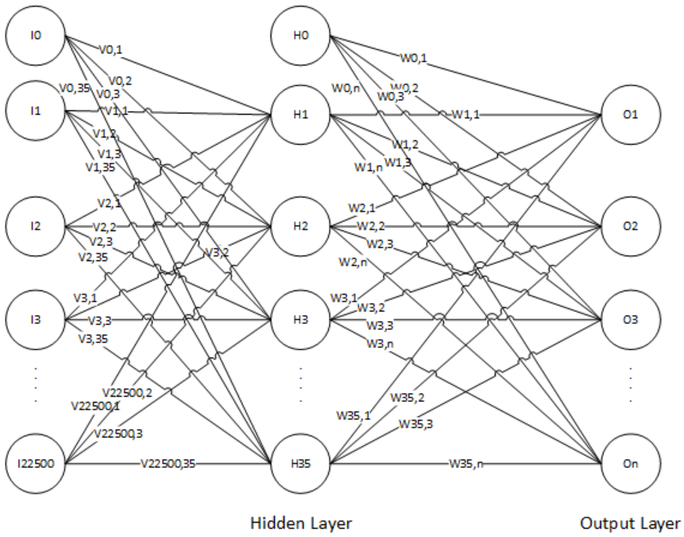

Figure. 1. ANN Architecture

6. Data Distribution
   A 2:1 ratio is maintained between training and testing data. Specifically, the dataset comprises 61 training images and 30 testing images.

4. Implementation

During the trial phase, there were 91 images consisting of five types of flowers. Specifically, there were 16 images of Pink Roses, 19 of Frangipani, 16 of Purple Trumpet flowers, 22 of Plumeria, and 18 of Red Roses. From each flower type, six images were selected as testing data. This resulted in 30 images used for testing and 61 for training.

The initial weights used in the experiment were randomized for every trial. This decision was based on previous tests which showed that artificial neural networks with randomized initial weights had better accuracy than those with static initial weights. The training parameters for the neural network were set as follows: a learning rate of 0.001, a momentum rate of 0.01, and a threshold of 0.451. The choice of a 0.001 learning rate was due to the fact that a smaller learning rate increases the chances of the neural network's weights achieving maximum accuracy. However, a drawback of a smaller learning rate is that it takes a longer time to train the neural network. The momentum rate was set at 0.01 for the same reasoning behind the learning rate choice. The threshold of 0.451 was chosen because the initial experiment set a starting threshold close to 0.5.

After three days of training, the data from the experiment revealed that a hidden node count of 35 was optimal. This was because the accuracy obtained from the experiment was 80%, and it was the highest accuracy achieved among smaller hidden nodes.

5. Discussion

The advancement in machine learning, especially in neural networks, has enabled a myriad of applications across diverse domains. The development of an Android-based flower classification application, as discussed, is a testament to this. A few key aspects from the given results deserve attention:

- Data Division: The experiment leveraged 91 images from five distinct flower types. The choice of selecting only six images from each type as testing data, thereby using the majority (61 images) for training, underscores the importance of having a robust training dataset to train the neural model comprehensively.
- Randomized Initial Weights: Traditionally, neural networks can be sensitive to the initial choice of weights. The decision to employ randomized weights every trial, and its resultant improved accuracy over static initial weights, could be attributed to the network's capability to escape local minima and explore the solution space more efficiently. This approach, although not universally optimal for every problem, seems

to have been very effective for this specific flower classification task.
- Learning Rate and Momentum: The choice of a smaller learning rate, 0.001, aligns with the idea that smaller steps can lead the network towards a more precise solution. However, as highlighted, it comes with the trade-off of longer training times. The addition of momentum, set at 0.01, helps accelerate the network's convergence by adding a fraction of the previous weight update to the current one. This ensures a smoother and possibly faster approach to the solution, especially in valleys in the loss landscape.
- Threshold Setting: The chosen threshold value of 0.451, close to 0.5, is intriguing. While the reason behind this specific threshold isn't detailed, it would be interesting to explore how variations in this value might impact the network's performance.
- Optimal Hidden Node Count: After the rigorous training process, determining that 35 hidden nodes yielded the highest accuracy is noteworthy. While the model achieved an impressive 80% accuracy, one might wonder if further fine-tuning or augmenting the training data could lead to even better results.
- Practical Implications: The application, once deployed on Android devices, could serve botanists, flower enthusiasts, or even common users in recognizing and classifying flowers. The real-world effectiveness of the app would be contingent upon its performance with diverse and unseen data, beyond the 91 images used during development.
- Future Enhancements: It would be worthwhile to consider augmenting the dataset with more images, both in terms of quantity and variety. Incorporating images taken under different conditions—like varying lighting, angles, or seasons—might make the model more robust. Also, exploring deeper architectures or other advanced neural network techniques could further improve accuracy.

4. Conclusion

From the research conducted, it can be concluded that the flower identification application using the artificial neural network with the backpropagation method and edge detection on Android has been developed using a server-client system. This application can recognize flowers that have been trained with an accuracy rate of 80%. This level of accuracy is likely influenced by the limited dataset and the potential for significant variations in patterns from images of similar flowers. To further expand on the findings:
- Server-Client System: The decision to employ a server-client system for the application suggests a centralized model where the computational-intensive process of neural network predictions can be handled by the server, thus alleviating the client (or user's Android device) from the heavy processing. This design choice might enhance the application's scalability and performance on diverse Android devices (Gupta et al., 2023).
- Accuracy Rate: An accuracy of 80% is commendable, especially given the inherent challenges in flower pattern recognition due to the intricacies and nuances in flower designs. However, for practical, real-world applications, a higher accuracy would be desirable.
- Dataset Limitations: The limited dataset is a constraint that might have impacted the model's performance. A more extensive dataset that encompasses a wide variety of flower images, captured under different conditions and from various angles, could potentially enhance the model's accuracy and robustness.
- Pattern Variations: The observation that there are significant variations in patterns even among images of similar flowers underscores the complexity of the task. It suggests the need for a more robust preprocessing or feature extraction method, which could aid the neural network in discerning and distinguishing between nuanced patterns more effectively.
- In light of these findings, future work could focus on dataset augmentation, refining the feature extraction process, and perhaps exploring more sophisticated neural network architectures or hybrid models to further improve the application's accuracy and reliability.

Author Contributions: All authors contributed to this research.

Funding: Not applicable.

Conflict of Interest: The authors declare no conflict of interest.

Informed Consent Statement/Ethics Approval: Not applicable.

Acknowledgments: Thank you to Universitas Multimedia Nusantara for providing this research facility.

References

Aznan, A. A., Ruslan, R., Rukunudin, I. H., Azizan, F. A., & Hashim, A. Y. (2017). Rice Seed Varieties Identification based on Extracted Colour Features Using Image Processing and Artificial Neural Network (ANN). *IJASEIT*, *7*(6), 2220–2225.

Gupta, A., Sevak, H., Gupta, H., & Solanki, R. K. (2023). Swiggy Genie Clone Application. *Int. J. of Aquatic Science*, *14*(1), 280–287.

Hiary, H., Saadeh, H., Saadeh, M., & Yaqub, M. (2018). Flower classification using deep convolutional neural networks. *IET Computer Vision*, *12*(6), 855–862.

Jaiswal, M. (2018). Android the mobile operating system and architecture. *Manishaben Jaiswal," ANDROID THE MOBILE OPERATING SYSTEM AND ARCHITECTURE", International Journal of Creative Research Thoughts (IJCRT), ISSN*, 2320–2882.

Kusnadi, A., Pane, I. Z., Overbeek, M. V., & Prasetya, S. G. (2023). Face Recognition Accuracy Improving Using Gray Level Co-occurrence Matrix Selection Feature Algorithm. *2023 International Conference on Smart Computing and Application (ICSCA)*, 1–6.

Kusnadi, A., Wella, R. W., Pane, I. Z., & Fasa, M. I. (2022). Multi-Instance Face Recognition System Using Pca And Ann. *Nveo-Natural Volatiles & Essential Oils Journal| NVEO*, 710–718.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., & Kolesnikov, A. (2020). The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, *128*(7), 1956–1981.

Marmanis, D., Schindler, K., Wegner, J. D., Galliani, S., Datcu, M., & Stilla, U. (2018). Classification with an edge: Improving semantic image segmentation with boundary detection. *ISPRS Journal of Photogrammetry and Remote Sensing*, *135*, 158–172.

Meester, M. J., & Baslamisli, A. S. (2022). SAR image edge detection: review and benchmark experiments. *International Journal of Remote Sensing*, *43*(14), 5372–5438.

Nwadiugwu, M. C. (2020). Neural networks, artificial intelligence and the computational brain. *ArXiv Preprint ArXiv:2101.08635*.

Rao, Y., Zhao, W., Zhu, Z., Lu, J., & Zhou, J. (2021). Global filter networks for image classification. *Advances in Neural Information Processing Systems*, *34*, 980–993.

Shrestha, K. K., Bhattarai, S., & Bhandari, P. (2018). *Handbook of Flowering Plants of Nepal (Vol. 1 Gymnosperms and Angiosperms: Cycadaceae-Betulaceae)*. Scientific publishers.

Waldner, F., & Diakogiannis, F. I. (2020). Deep learning on edge: Extracting field boundaries from satellite images with a convolutional neural network. *Remote Sensing of Environment*, *245*, 111741.