King Saud University

College of Computer and Information Sciences

Department of Computer Science

CSC462: Machine Learning

Course project

Second semester 2025

Final report

Group 4

Section 46682

Instructor:

L. Rawabi Alwanin

Student Name	*Student ID*	*Participation*
Hissah Ibn Qurmulah	443200791	Related Work Model design Model Development and Training Data Preprocessing
Reem Alsuhaim	443200884	Related Work Model design Model Development and Training Data Preprocessing
Remas Almania	443201068	Related Work Model design Model Development and Training Comparison and Evaluation

Rose classification

I. **Introduction**

Proper classification is crucial since flowers are used extensively in the medicinal, cosmetic, and agricultural sectors. However, conventional approaches are labor-intensive and prone to human mistakes since they depend on morphological characteristics and require specialized knowledge. Flowers have many uses outside of aesthetics, such as in food production and medicine, as mentioned in [1], which makes effective classification methods more critical. By automating feature extraction and increasing accuracy, deep learning particularly Convolutional Neural Networks, or CNNs has revolutionized picture classification. CNNs are suitable for classifying flowers; Gurnani et al. [2] showed that they performed better than conventional feature-based techniques. Utilizing CNNs can make flower classification automated, more accurate, and scalable, which will help several sectors that depend on flower-based.

II. **Problem Definition**

We chose to create a model capable of classifying ***Roses*** from other flowers because roses were and remain a symbol among people to express their feelings. They are silent messages that represent emotions that language fails to capture. To achieve this, we will develop a Convolutional Neural Network (CNN) model for binary classification, specifically designed to distinguish roses from all other types of flowers.

III. **Dataset**

We collected the data after an extensive search, resulting in a dataset containing 100 images of roses and 25 images for each of tulips, dandelions, sunflowers, and daisies, to align with our goal, which is to create a model capable of distinguishing roses from other flowers. The images were gathered from search engines and social media platforms.

IV. **Related Work**

Flowers play an important role in the food chain and in maintaining biodiversity. Since there are many different species, manual identification can be challenging, even for experts. This is where deep learning comes in, as it helps extract important features such as color, shape, and texture. It can also classify flower images into different types using convolutional neural networks.

The researchers in this paper [3] relied on CNN techniques. They followed a transfer learning approach that consists of four main stages. The process starts with image preprocessing, where each image is resized and normalized to fit the model. Then, the model is trained using DenseNet121 with the fine-tuning technique. After that, the model is validated, and finally, it is tested to evaluate its performance. They relied on Google Colab with a Tesla K80 GPU unit to conduct the experiments and used the Oxford-102 dataset, which contains 8,228 images. Of these, 6,583 images were used for training, with the remaining images divided between validation and testing. It was observed that this model achieved high accuracy compared to other methods.

This paper [1] compares the performance of CNN and SVM in the task of flower classification using a dataset containing 400 images of roses, 400 of tulips, and 400 of asters. Initially, the images were enhanced through preprocessing and resized to 200×200 pixels. The data was then split into 80% for training and 20% for testing. A CNN model was built with 4 convolutional layers, each using a 3x3 filter. The first layer has 16 filters, while the second layer has 16,32 filters. The third layer has 64 filters, and the fourth layer has 128 filters. ReLU was used as the activation function for all convolutional layers, and Max Pooling with a 2×2 kernel was applied to reduce the image dimensions. To avoid overfitting, L2 Regularization and Dropout with a rate of 0.5 were applied. The model includes 3 dense layers, with the first and second layers having 256 neurons each, while the last layer has 128 neurons. Epochs were set to 20.The results showed that the CNN model outperformed SVM, achieving 91.6% in Precision and Accuracy, and 92% in Recall**.**

This study [2] by Gurnani et al. explored the use of CNNs for classifying floral species using the Oxford 102 Flower Dataset, which contains 8,189 images across 102 categories. The study compared two CNN architectures, GoogleNet and AlexNet, by fine-tuning them on the dataset. GoogleNet outperformed AlexNet, achieving a Top 1 accuracy of 47.15% and a Top 5 accuracy of 69.17%, demonstrating the advantages of deeper networks and improved regularization.It has shown CNNs a remarkable performance in image classification, replacing traditional methods that relied on handcrafted features like SIFT and HOG. By automating feature extraction, deep learning has significantly improved classification accuracy. The study shows the benefits of deep CNN architectures, showing that deeper networks with better regularization achieve higher accuracy. It also emphasized the effectiveness of transfer learning, where pre-trained models enhance performance when fine-tuned on domain-specific datasets. These findings confirm CNNs crucial role in automating feature extraction and achieving state-of-the-art classification results [2].

V. **Data Preprocessing**

To increase diversity and make the model more flexible in recognizing images, we applied several modifications such as zooming, rotation, and shifting the image to the left or right. We made sure that only one type of modification was applied to each image to maintain variety without excessive repetition. The modified images were then saved in a new folder, with each image placed in its corresponding class category. The augmentation process doubled the dataset size by 100%; for example, if a class originally had 100 images, it became approximately 200 after augmentation.

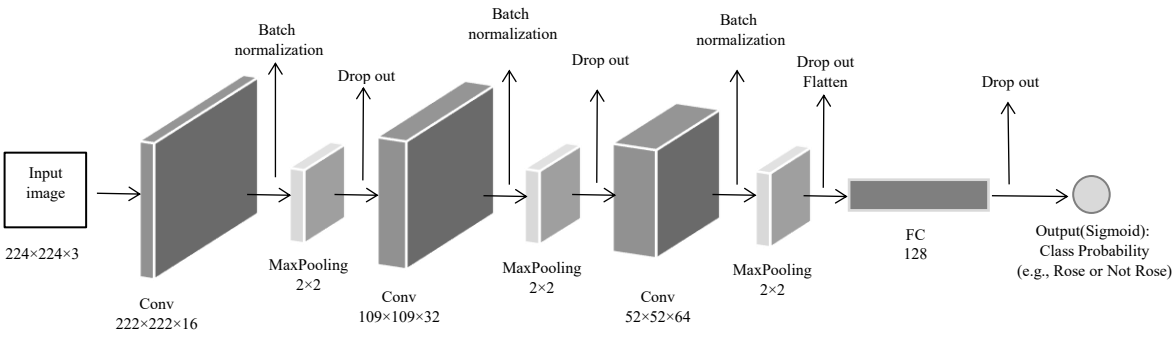

Figure 1 : illustration of the model structure

VI. **Model Design**

We built a Convolutional Neural Network (CNN) consisting of three convolutional layers followed by a dense layer. The input dimensions were 224x224x3, and the convolutional layers used filters of sizes 16, 32, and 64 respectively, with the ReLU activation function. Each convolutional layer was followed by a 2x2 MaxPooling layer, and we applied Batch Normalization to accelerate training and improve stability. The dense layer contained 128 neuron with a ReLU activation, and the final output layer used the sigmoid function to produce class probabilities.

After testing this initial architecture, we observed that the model suffered from overfitting. To address this, we added L2 regularization with a value of 0.001 in the first two convolutional layers and increased it to 0.005 in the third convolutional layer and the dense layer. We also applied a Dropout rate of 0.3 across all layers to reduce overfitting. (See Figure 1 for a visual representation.).

VII. **Model Development and Training**

The model was implemented and trained using TensorFlow and Keras. Input images were normalized and resized to match the expected dimensions. To further improve the model's generalization, we applied data augmentation, effectively doubling the dataset by applying single transformations such as zooming, rotation, or shifting on each image, and saving the results in class-specific folders.

The dataset was split into three sets using the hold-out method: 70% for training, 15% for validation, and 15% for testing. The batch size was set to 32 for the training set and 64 for both validation and test sets to maintain a balance between learning efficiency and memory usage.

Training was performed with a learning rate of 0.0001, and although we allowed a high number of epochs to maximize learning, EarlyStopping with a patience of 5 was used to halt training once performance plateaued.

Model evaluation was conducted using accuracy, precision, recall, and F1-score. Hyperparameter tuning was carried out by experimenting with various learning rates, dropout values, L2 regularization strengths, and decision thresholds. Through this, we found that the model favored the positive class at a threshold of 0.5, and the negative class at 0.6, so we adjusted the threshold to 0.53 to better balance precision and recall.

VIII. **Comparison and Evaluation**

Two models were developed and evaluated for binary flower classification: a custom Convolutional Neural Network (CNN) and a baseline model based on transfer learning using MobileNetV2 [4]. We use MobileNetV2 as baseline . It is for mobile and embedded vision applications, which makes it faster and less memory-intensive than deeper networks, which makes it perfect for smaller datasets like ours.

Both models were trained on the same balanced and augmented dataset. The custom CNN achieved 95% accuracy, 96.55% precision, and 93.33% recall, outperforming the baseline model which scored 91.67% accuracy, 93.10% precision, and 90.00% recall. (Shown on table 1) The confusion matrix further supports this, as the custom model misclassified only 3 images compared to 5 in the baseline. (Table 2) (Table 3)

Although the baseline model showed smoother validation curves with less fluctuation and a slightly smaller gap between training and validation loss, (See figure 2, 3) the custom CNN demonstrated better class separation and a more balanced prediction behavior, especially after adjusting the decision threshold. Overall, while MobileNetV2 is a powerful option for general use, the custom CNN was more effective in this task, showing better adaptation, faster convergence, and superior evaluation performance across multiple metrics.

Table 1 : *Evaluation Metrics*

	Precision	***Recall***	***F1-Score***	***Accuracy***
Our model	0.97	0.93	0.95	0.95
baseline	0.93	0.90	0.92	0.92

Table4 : *confusion matrix of our model*

Predicted Negative	***Predicted Positive***	
29	1	***Actual Negative***
2	28	***Actual Positive***

Table2 : *confusion matrix of baseline*

Predicted Negative	***Predicted Positive***	
28	2	***Actual Negative***
3	27	***Actual Positive***

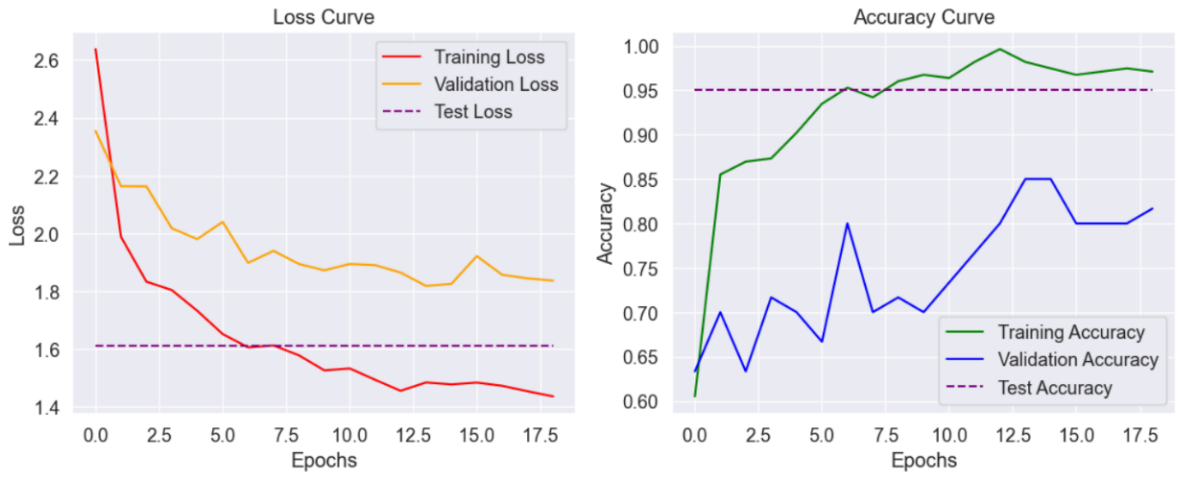

Figure2 : loss curve and accuracy curve of our model

Figure3 : loss curve and accuracy curve of baseline model

IX. **Discussion**

	Our model	**Baseline**
Aspect	**Custom CNN**	**MobileNetV2**
Architecture	3 Conv layers with L2 regularization, batch norm, and dropout	MobileNetV2 that has already been trained with dense layers using L2 regularization and dropout
Model Size	Less parameters and lightweight	Bigger, more profound, and pretrained
Training Approach	trained from start	Using a frozen basis to transfer learning
Generalization	Minimal overfitting and strong generalization	A little less consistent validation result

Training Stability	Stable and fast convergence during epochs	Smooth but slower due to transfer learning freeze
Computation Cost	Minimal memory usage and reduced training duration	takes longer and more resources.

The model shows strong performance, particularly on the test set. These results indicate that the model is highly effective in distinguishing between the two classes, with minimal false positives and false negatives. It showed decent performance , suggesting good generalization.

However, there are some limitations. The accuracy and loss curves revealed a noticeable gap between the training and validation performance, (See figure 2) indicating potential overfitting. Additionally, the validation accuracy showed some fluctuation across epochs, which may be due to the relatively small dataset size (400 images). Despite augmentation, this limited data volume can affect model stability and robustness.

For future improvement, collecting more real-world images could significantly enhance model performance and reduce overfitting. Applying stronger regularization techniques may also help close the gap between training and validation performance.

X. **Conclusion**

In conclusion, the model showed strong potential for accurate flower classification. Despite some limitations, the results demonstrate the potential of CNN-based models in real-world applications. With further tuning and a larger dataset, it can support broader applications in agriculture and botany.

References

[1] A. Peryanto, A. Yudhana, and R. Umar, “Convolutional neural network and support vector machine in classification of flower images,” *Peryanto | Khazanah Informatika : Jurnal Ilmu Komputer Dan Informatika*, Mar. 03, 2022. https://journals.ums.ac.id/index.php/khif/article/view/15531/7387

[2] A. Gurnani, V. Mavani, V. Gajjar, and Y. Khandhediya, “Flower Categorization using Deep Convolutional Neural Networks,” *arXiv.org*, 2017. https://arxiv.org/abs/1708.03763

[3] N. Alipour, O. Tarkhaneh, M. Awrangjeb, and H. Tian, “Flower Image Classification Using Deep Convolutional Neural Network,” *2021 7th International Conference on Web Research (ICWR)*, May 2021, doi:https://doi.org/10.1109/icwr51868.2021.9443129.

[4] G. , Z. M. , C. B. , K. D. , W. W. , W. T. , A. M. , & A. H. Howard, “Efficient Convolutional Neural Networks for Mobile Vision Applications,” arXiv preprint, 2017.