Deep Learning-Based Flower Classification

Group 17

1. Estellyn Hoo Weay 1211300202
2. Lee Tian Xin 1211301744
3. Ho Xin Yong Cinee 1201303155
4. Chan Kah Kei 1211307001

Introduction

- Flower classification is a complex problem in computer vision due to variability in images, such as lighting, background, colours, and angles.
- Traditional methods may not reliably and accurately classify flower types.
- Motivations:
  - To contribute to advancements in computer vision
    - Potential for developing robust algorithms that could also enhance capabilities in other fields.
  - To improve agricultural applications
    - Crop management, pest control, and pollination strategies.
  - To support research and conservation efforts
    - For mapping the geographic distributions of flowers, pinpointing areas rich in biodiversity, and monitoring changes in plant populations over time.

Introduction

- Objectives:
  - To develop a deep learning model that can accurately and reliably classify different types of flowers despite variations in lighting, background, colour, and angle.
  - To evaluate and compare the performance of deep learning models and identify the most effective model based on various performance metrics.

Literature review

- **Challenge:**
  - Wide array of flower classes that share similar shapes, appearances or background elements.
  - Traditional methods rely on features extracted from segmented flower regions only, limiting the ability to capture the full spectrum of variability in flower images.

- **Performance metric:**
  - Accuracy

- **Most studies used either deep learning alone or in combination with machine learning methods.**

- **Results:**
  - Deep learning methods achieved high accuracy, often exceeding 87%.
  - Machine learning methods show more variable accuracy (74.59%-98.5%).

- **5 Flower Types Classification** dataset from Kaggle
  - Lily, Lotus, Sunflower, Orchid, and Tulip
  - 1,000 images for each flower type

- The flower images, even of the same type, differ in lighting, background, colours, and angles

Lily

Lotus

- The flower images, even of the same type, differ in lighting, background, colours, and angles

Sunflower

Orchid

Tulip

Load Image

- Extract images from compressed file.
- Verify extraction to ensure all images are loaded.

Data Split

- Split dataset
  - Train: 64%
  - Validation: 16%
  - Test: 20%
- Set random state to 42 for reproducibility.

Create DataFrame

- Organize and label flower images.
- Construct file paths and store with labels.
- Create a DataFrame with columns for file paths and labels.

Resize and Normalize

- Resize images to 224x224 pixels with three color channels.
- Batch images in groups of 32.
- Normalize image data to range [0,1].

Convolutional Neural Networks (CNNs)

- Excels in image recognition, classification, and object detection.
- Components:
  - Convolutional layers to detect features.
  - Pooling layers to reduce dimensionality.
  - Fully connected layers to classify images.

DenseNet-121

- A DenseNet with 121 layers.
- Dense connections enhance feature reuse and gradient flow.
- Transfer Learning: Pre-trained on ImageNet

InceptionV3

- Uses parallel convolutions of varying sizes.
- Effectively learn complex patterns in images while maintaining computational efficiency.
- Transfer Learning: Pre-trained on ImageNet

Custom CNN

- Custom architecture designed for image classification.
- Includes convolutional layers, max pooling, dropout layers, batch normalization, and fully connected layers.

Experiment Setup

- Batch Size: 32
- Epochs: 20
- Loss Function: Categorical Cross-Entropy
- Optimizers: Adam and SGD
- Early Stopping: Validation loss with patience of 7 epochs
- Packages and Libraries: TensorFlow and Keras

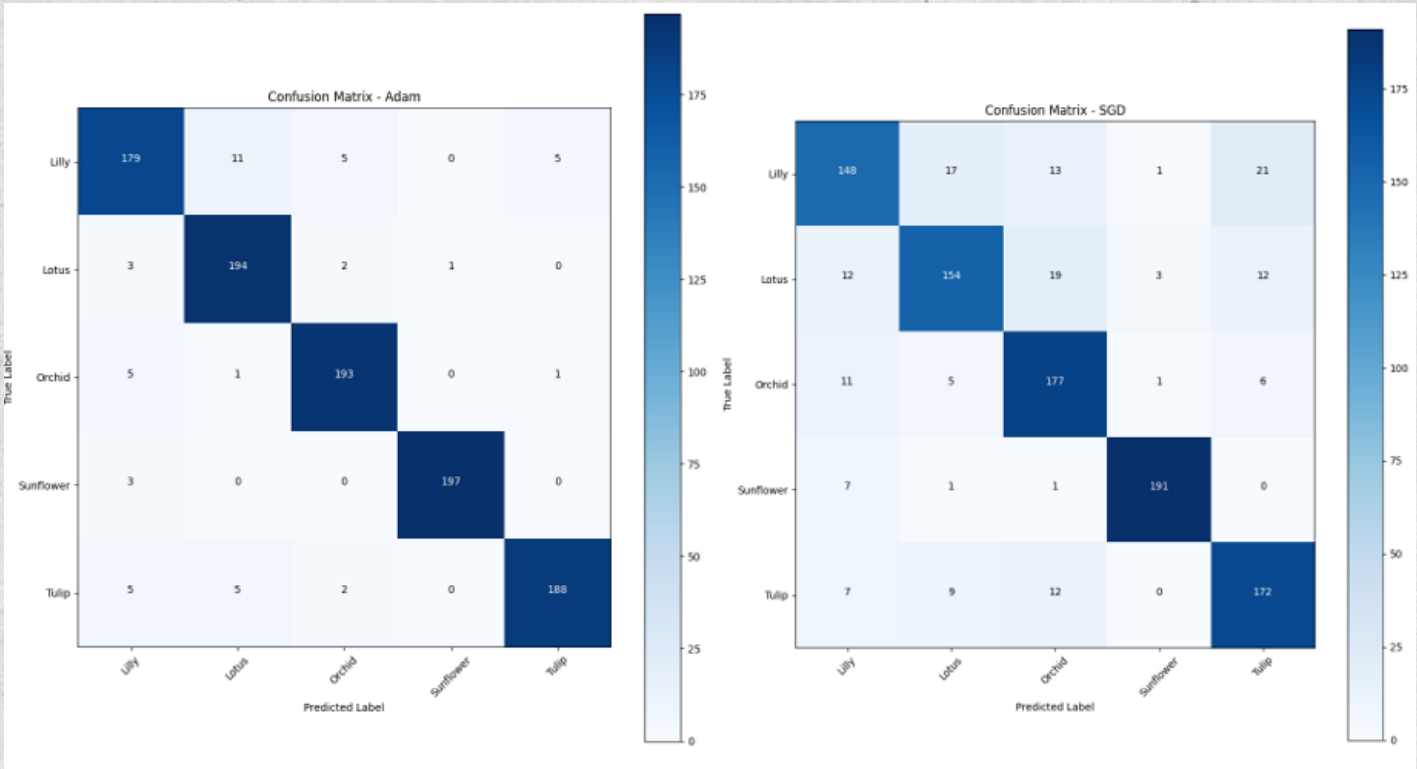
Performance Metrics

- **Accuracy:** Proportion of true results among examined cases.
- **Precision:** Ability to correctly identify flowers of a specific type without labeling unrelated flowers as such.
- **Recall:** Ability to correctly identify all instances of a particular flower type.
- **F1 Score:** Harmonic mean of precision and recall to balance false positives and negatives.
- **Confusion Matrix:** A table used to evaluate the performance of a classification model by showing the true versus predicted classifications.
- **Prediction Time**: Time taken by the model to make a prediction on new data.
- **Parameter Size**: Total number of parameters (trainable and non-trainable)

Experimental Results

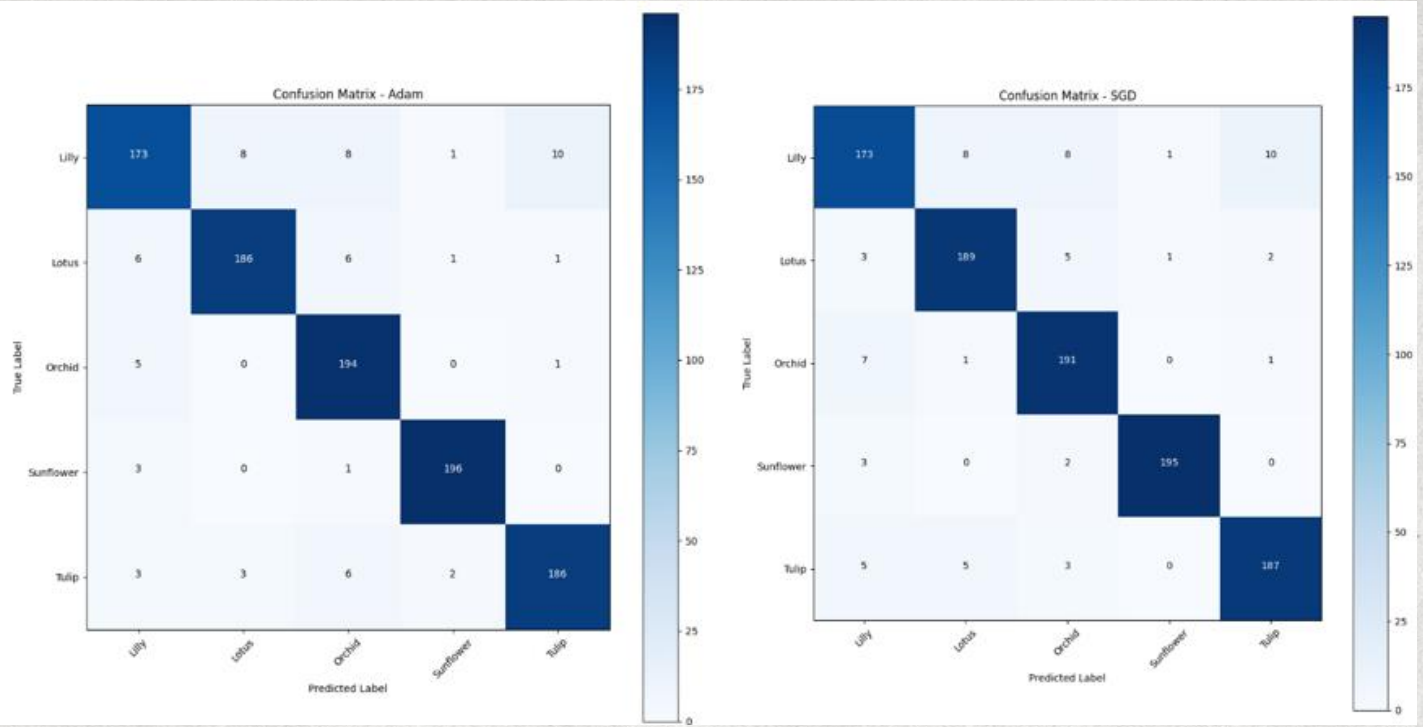
Model	Opt.	Accuracy	Precision	Recall	F1 score	Prediction Time(s)	Total Parameter size (MB)
DenseNet-121	Adam	0.95	0.95	0.95	0.95	5.82	75.84
	SGD	0.84	0.84	0.84	0.84	6.96	
InceptionV3	Adam	0.94	0.94	0.93	0.93	6.88	142.24
	SGD	0.94	0.94	0.94	0.93	5.01	
Custom CNN	Adam	0.68	0.69	0.68	0.68	4.74	74.74
	SGD	0.57	0.69	0.57	0.55	3.72	

Confusion Matrix of DenseNet-121



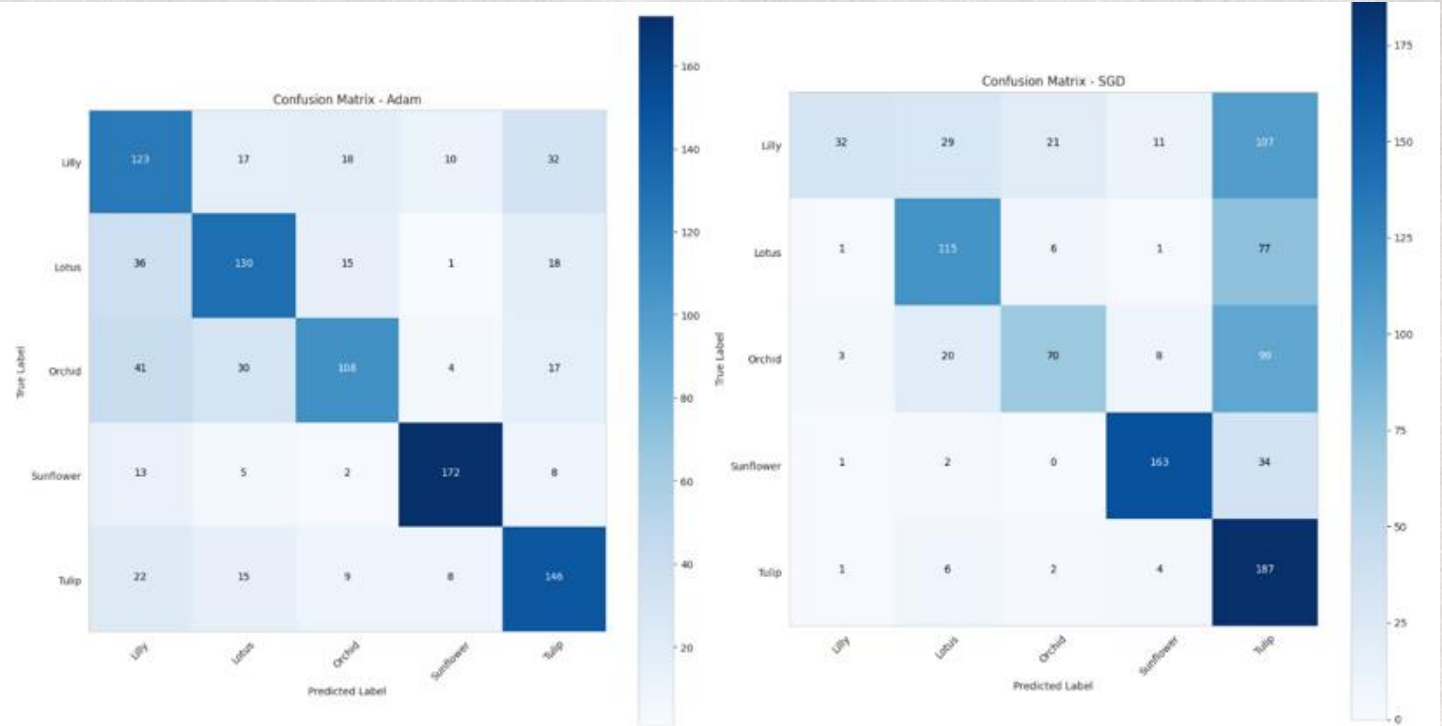
DenseNet-121 with Adam optimizer performed better than SGD (higher number of correct predictions for all flower types), while SGD model encountered difficulty in classifying Lily and Lotus.

Confusion Matrix of InceptionV3



InceptionV3 performed well with both Adam and SGD, although in both configurations, it occasionally confused Lily with Lotus, Orchid, and Tulip

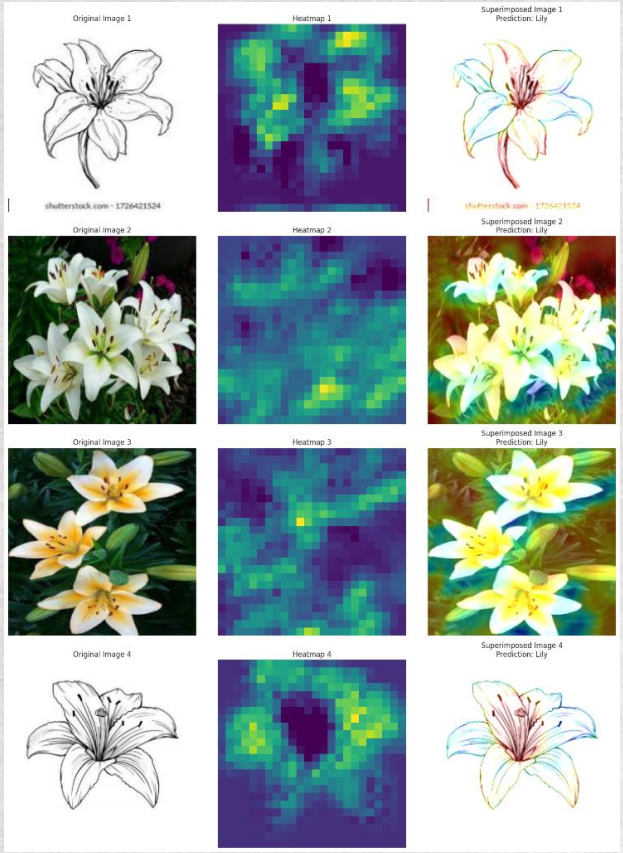
Confusion Matrix of custom CNN



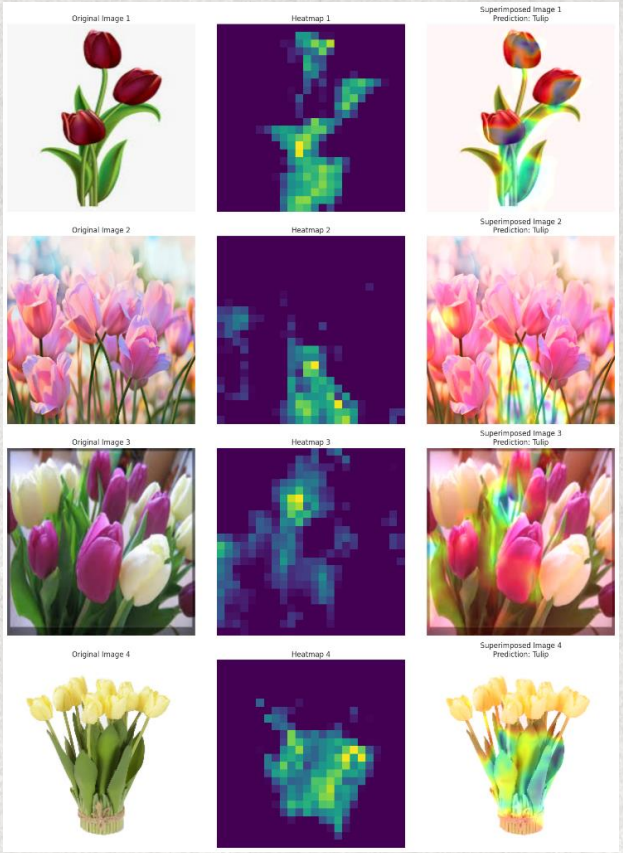
Adam model clearly outperformed the SGD model. SGD had issues with distinguishing Lily and Orchid since most of the predictions wrongly classified these two flower types to Tulip.

Grad-Cam of Custom CNN SGD

Correct Prediction of Lily

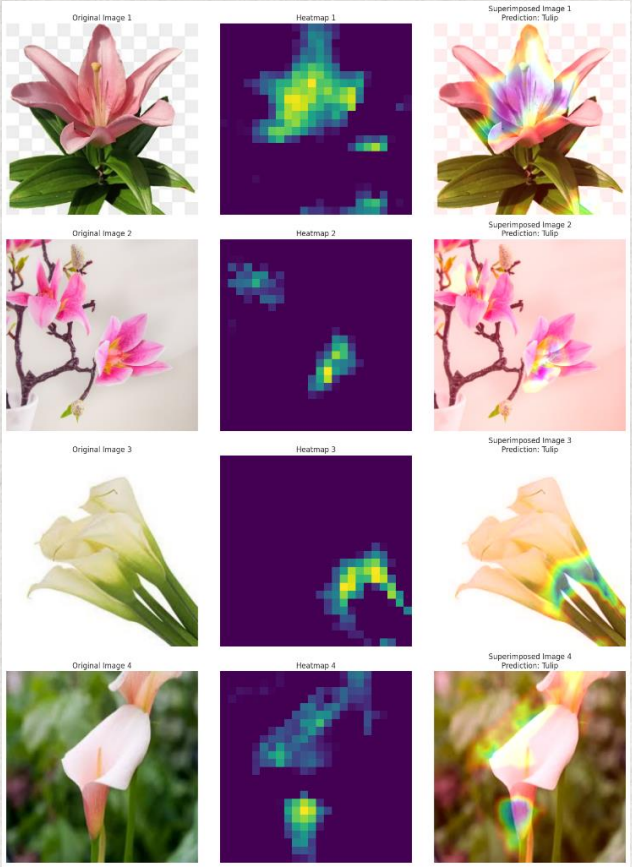


Correct Prediction of Tulip



Grad-Cam of Custom CNN SGD

Misclassification of Lily to Tulip



- The model failed to capture the complete petals of Lily, leading to incomplete feature extraction and misclassification.
- In some of the misclassified images, Lily petals resemble those of Tulip, especially the Calla Lily.

Conclusion

- DenseNet-121 with Adam emerged as the best-performing model with an accuracy of 95%
- Adam outperformed SGD when used as an optimization algorithm with DenseNet-121 and custom CNN
- InceptionV3 achieved in same performance for 2 optimizers.

Future Improvement

- Using advanced architectures capable of capturing finer details and improving feature differentiation
- Applying sophisticated data augmentation techniques and utilizing more diverse datasets