UNIVERSITY PARTNER

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

Artificial Intelligence and Machine Learning (6CS012)

Image Classification of Flowers Using Convolutional Neural Networks and Transfer Learning

Student Id	: 2358479
Student Name	: Umesh Gautam
Group	: L6CG7
Module Leader	: Mr. Siman Giri
Instructor	: Ms. Durga Pokhrel
Submitted on	: 18 May, 2025

Table of Contents

1. Abstract:

This project explores the application of Convolutional Neural Networks (CNN) to classify flower images into five categories: Daisy, Dandelion, Rose, Sunflower and Tulip. The study begins with a baseline CNN consisting of three convolutional layers and regularized CNN models to enhance performance. Various techniques such as batch normalization, dropout and L2 regularization were implemented to address overfit. Two optimizers, Adams and SGD were compared to evaluate their impact on model accuracy and convergence speed. In addition, transfer learning was used using a pre-trained VGG16 model with frozen convolutional layers and custom dense layers. The model was evaluated using training/validation accuracy, loss curves, classification reports and confusion matrices. In models, the learning method for transfer with VGG16 achieved the highest verification accuracy of 82.15% to improve both the baseline and deeper models. This study highlights the effectiveness of transfer learning and regularization to improve the CNN performance of real -world classification features.

2. Introduction

Flower image classification is a critical computer vision task with many practical applications in real-world scenarios such as plant identification, medical diagnosis, and automated surveillance. This work aims to classify flower images into five categories: daisy, dandelion, rose, sunflower, and tulip. Accurate flower classification can help in botanical education, garden management and decision-making of agriculture. Deep learning, especially Convolutional neural networks (CNN), has revolutionized image classification by learning directly hierarchical features from raw images without manual functional technique. CNN uses layers such as convolution, pooling and dense connections for extracting and processes of special functions.

The study detects and compares the performance of three different models: a baseline CNN, a deep CNN with regularization techniques such as dropout, batch normalization and L2 weight decay, and a transfer learning model using pre-trained VGG16 architecture. In addition, two optimizers - Adam and SGD are evaluated to assess dynamics and model convergence. The model is analyzed based on accuracy, loss curves, classification reports and confusion matrix. The goal is to decide which architecture and configuration supports the best accurate and strong flower classification.

The key points covered in the report are:

Problem Statement: Classifying flower images into five different categories.

The importance of the real world: Helps in plant identification of facilities, educational equipment and agricultural applications.

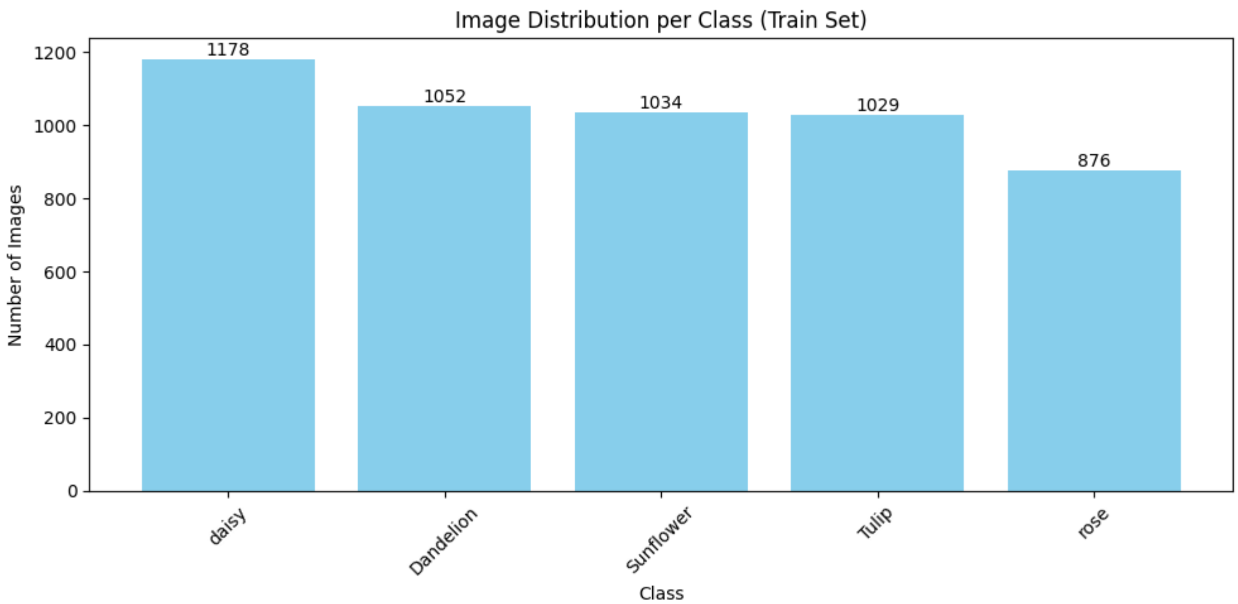
Background of deep learning: CNN automatically learns spatial hierarchies from images by using layers such as convolutional, pooling and activation functions.

Prior Work & Scope: The project compares three models:

- Baseline CNN
- Deep CNN and regularization
- Transfer learning with VGG16. The evaluation is dependent on classification accuracy, training curve, confusion matrix, and training time.

3. Dataset

The dataset used in this project is a custom flower data set with five categories: Daisy, Dandelion, Rose, Sunflower and Tulip. The basic data sets were obtained from Kaggle, a popular platform for science competitions and resources (Sleam, 2024). Many images were large watermarks or logos, which were removed manually to ensure pure visual input for exercise. In addition, new flower images were collected manually from open sources to increase diversity and class balance.



The dataset originally had random image size, which was standardized for 180x180 pixels using preprocessing techniques implemented through the ImageDataGenerator. This ensured uniform input size for CNN models.

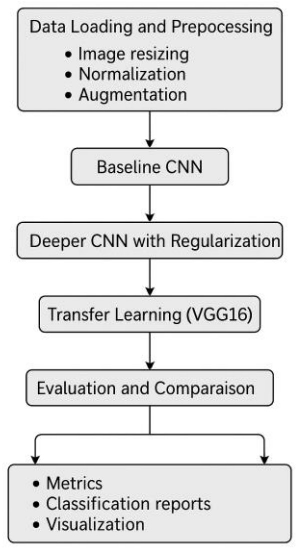
Preprocessing steps included:

- Change the size of all images of 180x180 pixels
- Normalization by scaling pixel values to the [0, 1] range
- Data augmentation using:
    - Rotation
    - Width and height shifts
    - Zooming

- Dataset Structure and Preparation:
  - Source: Custom dataset was retrieved from Kaggle and expanded manually.
  - Classes: Five (daisy, dandelion, rose, sunflower, tulip)
  - Image calculation: ~ 5169 images in all classes
  - Image resolution: 180x180 pixels
- Challenges Faced:
  - Manual cleaning of watermarks and presence of logo
  - Further pictures are required to balance classes
  - Original image variability in dimensions and quality
  - Preprocessing to normalize and increase image data

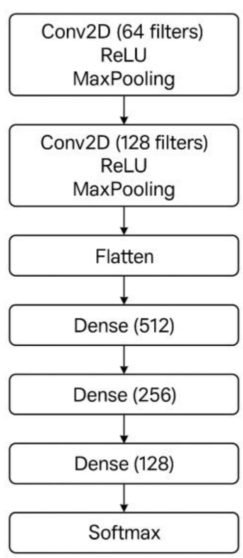
4. Methodology

To conduct a thorough evaluation of CNN-based image classification, the project followed a structured function. The workflow includes data preparation, preprocessing, model construction, training, evaluation and visualization. The dataset used includes flower images divided into training, verification and test sets. Image augmentation techniques such as rotation, flipping, zooming and changing were used to improve generalization. (Shorten, 2019)



Three models used were:

Baseline CNN: A simple architecture with three conviction teams after dense layers. Filter sizes (32, 64, 128), each followed by ReLU activation and MaxPooling.



Deeper CNN: To prevent the overfit of a more complex model that is overfit with additional conventional layers, dropout layers, batch normalization and L2 regularization. The architecture follows a pattern of Conv2D → BatchNorm → Dropout → Conv2D → MaxPooling, repeated with increasing depth.

Transfer learning with VGG16: A pre-trained VGG16 model from the ImageNet Frozen Convolutional layers for classification and custom dense layers.

Custom dense layers were added on top:

Flatten → Dense(512, ReLU) → Dropout(0.5) → Dense(5, Softmax)

Each model was trained using a training kit and validated using a holdout verification set. Early Stopping was used to avoid overfitting and improve convergence. Performance matrix such as training/verification accuracy, loss and classification reports were collected. The model was compared to accuracy, confusion matrix, training time and ROC curves.

Both Adam and SGD optimizer were tested to compare their impact on model convergence and accuracy. Visualization tools such as matplotlib and seaborn were used to plot confusion matrices along with accuracy and loss.

5. Experiments and Results:

CNN's behavior can vary greatly depending on architectural design, training management and adaptation methods. In the project, three CNN-based approaches were compared: a simple baseline CNN, a deep common CNN and a transfer learning VGG16-based model. To compare the model widely, a series of experiments were held on display measurements, calculation efficiency and training dynamics. Each model was trained on the same flower image dataset with equally pre-preparing and growth parameters for equal comparison. Experiments included the depth of different models, optimization methods and observing the dynamics of training in a dialect, to understand how the design options affect accuracy, training duration and generalization.

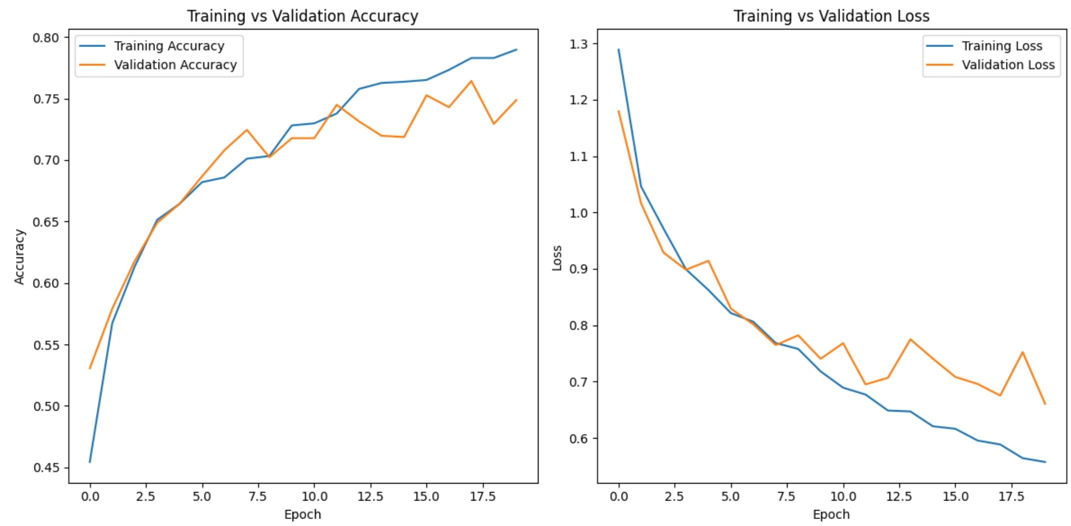
5.1 Baseline vs. Deep Architecture

- The Baseline model received 74.88% confirmation accuracy, while deep regularly reached 66.53% with CNN Adam Optimizer.

- Although deep models started regularization to reduce overfitting, it suffered from slow convergence and required more attitude.

- The Baseline model had a small architecture and was trained quickly (~ 534 seconds), while deep models took ~ 582 seconds.

- Training and verification decline showed that the Baseline model is better generalized with less ages.



The training vs. validation accuracy and loss curves for the baseline CNN model are shown in the figure above. The model shows a steady increase in accuracy and loss of losses in 20 epoch, with careful tracking of performance for verification performance. This indicates effective teaching without important overheating. Towards the end of the training, the Baseline model reached about 75% verification accuracy, which provides a solid goal to compare deep architecture and move learning strategies.

5.2 Computational Efficiency

- All training was done on Google Colab using GPU acceleration (Nvidia Tesla T4).

- The baseline model required low computation due to its small size.

- The deep model consumed more memory due to several layers, batch normalization and dropout.

- Transfer learning with VGG16, despite having the freezing layers, ii required long -term training time due to its complexity (~ 1080 seconds)

```
130/130 ━━━━━━━━━━━━━━━━━━━━ 51s 391ms/step - accuracy: 0.6895 - loss: 1.1861 - val_accuracy: 0.6062 - val_loss: 1.3432
Epoch 9/10
130/130 ━━━━━━━━━━━━━━━━━━━━ 50s 388ms/step - accuracy: 0.6876 - loss: 1.1911 - val_accuracy: 0.6101 - val_loss: 1.3103
Epoch 10/10
130/130 ━━━━━━━━━━━━━━━━━━━━ 93s 472ms/step - accuracy: 0.6962 - loss: 1.2041 - val_accuracy: 0.6838 - val_loss: 1.1548
Deeper Model Training Time: 582.02 seconds
Training Time Comparison:
Baseline Model: 534.41 seconds
Deeper Model: 582.02 seconds
```

5.3 Training with Different Optimizers

- The deep CNN was trained with both Adam and SGD Optimizer.
- Adam showed better convergence speed and gained high verification accuracy (66.53%) compared to SGD (44.61%).
- Training loss with Adam continuously decreased, while SGD showed slow and unstable behavior.

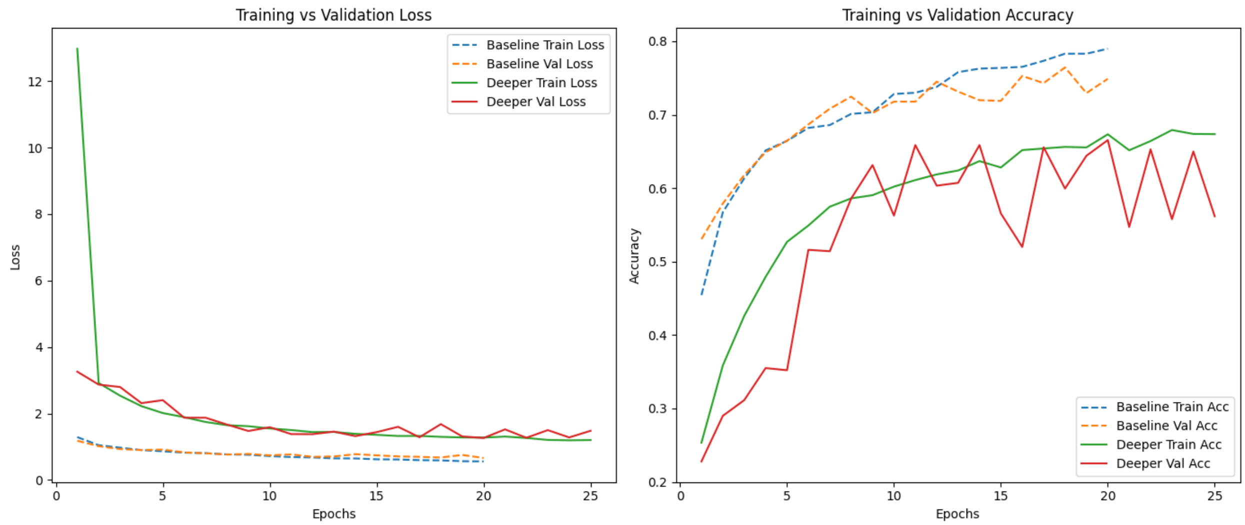
- Result:

```
Adam Optimizer - Best Validation Accuracy: 0.6653734445571899
SGD Optimizer  - Best Validation Accuracy: 0.44616878032684326
```

5.4 Challenges in Training

- Despite the use of dropout and L2 regularization, deep models were seen overfit.
- It was necessary to balance the power of improvement to avoid underfiting.
- The depth of the model and the time of training depends on the basic choice, which varies greatly.
- Model increased calculation costs with complexity; Regular and VGG16 models require more resources.

The figure shows a visual comparison of the results of prediction from three models-on the same set of images of baseline CNN, fine-tuned (transfer learning) and deep CNN flowers. Each column represents the predictions of a model, including the approximate class and its trust points. The fine-tuned model continuously achieves the highest accuracy and confidence and identifies all the samples correctly but it is misclassifying the image in the figure. Opposite reservations the dark model has many flowers, and the baseline model reflects moderate performance, which reveals better reliability of transmission learning.



The deep CNN model showed overfitting despite the use of dropout, batch normalization and L2 regularization. While training accuracy continuously improved, the confirmation accuracy was inconsistent and low. The gap between training and verification losses has learned the model well but struggled to normalize new data.

6. Fine-Tuning or Transfer Learning

For this project, we used VGG16, which was pre-trained on the ImageNet dataset by a widely recognized deeply conventional neural network. We used a functional approach, where all the interconnection layers were frozen, and only the recently added dense model were trained. This strategy allowed the model to take advantage of high -level functions learned from millions of images, reduce training time and improve generalization.

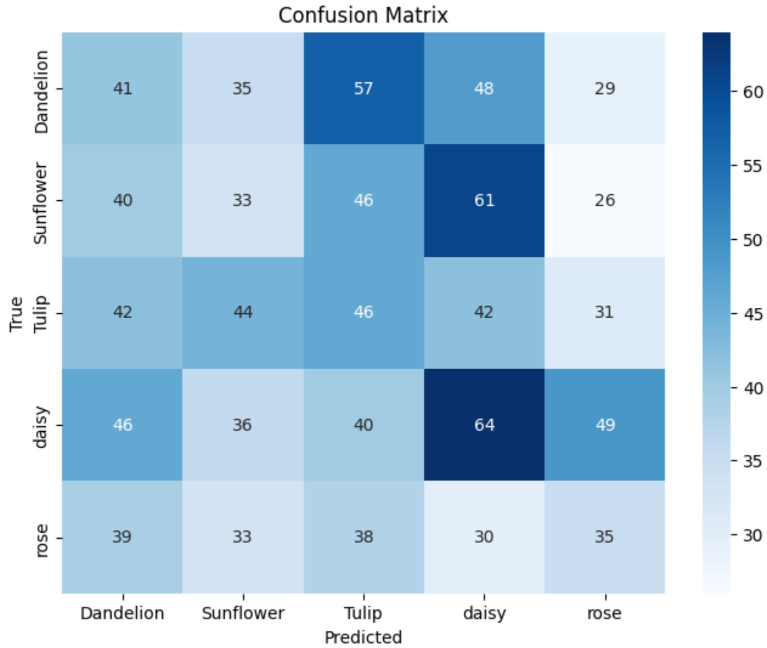
Architecture added a flatten layer, followed by dense (512, ReLU), a dropout (0.5) layer to prevent overfitting, and a final Dense (5, softmax) output team for classification.

Performance Comparison:

- Baseline CNN achieved ~ 74.88% validation accuracy.
- The deep CNN with regularization reached ~ 66.54%.
- The transfer of two learning with VGG16 improved them both and reached an accuracy of 82.15%.

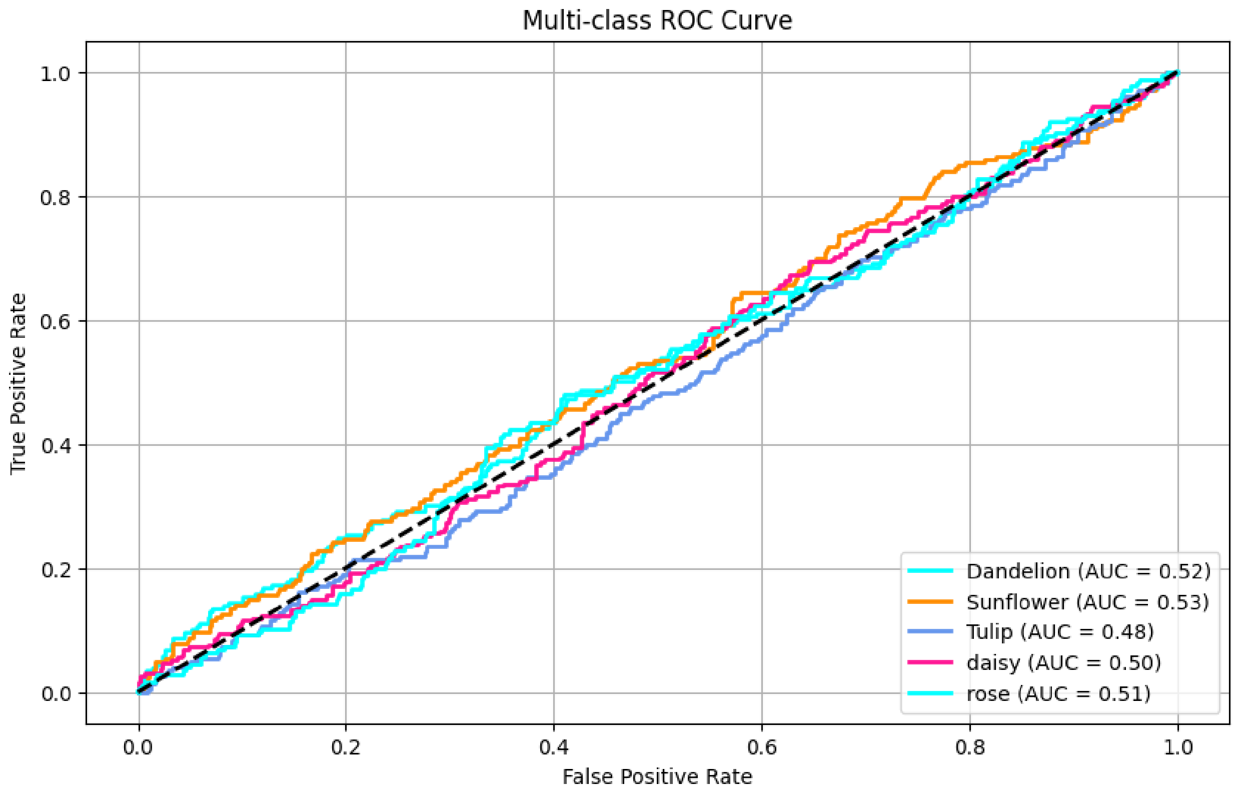
This shows the advantage of using a pre-informed model, especially on relatively small or noisy data sets, where scratches can lead to overfitting or poor generalization.

7. Experiments and results:



The above figure shows confusion matrix for VGG16 Transfer learning models for flower classification work. Each line represents the real class, and each column represents the approximate class. The diagonal values indicate correct predictions, while off-diagonal values indicate spontaneous abortion. The model performs best on the Daisy category, with 64 correct predictions, and shows relatively low accuracy of roses and dandelion classes. This visualization reveals the strength of the model in recognizing some types of flowering and reveals the illusion between the visually similar classes as tulip and dandelion. Matrix complements accuracy measurements by offering a deep classic performance insight.

This figure shows examples of predictions made by the Model of Learning (VGG16) transfer of unseen test images. Each image is marked with confidence score for the approximate class and model. The model correctly classified all five flower types - Daisy, Dandelion, Rose, Sunflower and Tulip - with high self -confidence, more than 94% in each case. This shows the model's strong capacity for normalization and reveals the advantage of taking advantage of pre -trained conversion teams from VGG16, which removes the rich visual functions required for accurate classification.



This figure represents the Receiver Operating Characteristic (ROC) for each flower class using transfer learning models. The area below the curve (the AUC) value is close to 0.5 for all classes Dandelion (0.52), sunflower (0.53), tulip (0.48), Daisy (0.50) and Rose (0.51) indicating the limited discriminatory capacity of models under the one-vs-rest classification. This suggests that although the general accuracy is high, the model may struggle with confident classes from trust during ROC evaluation, possibly due to class imbalance or equality in visual features in categories.

8. Conclusion and Future Work

The project compared baseline CNN, a deep common CNN and a VGG 16-based transmission learning model for flower classification. The transfer learning model performed best by achieving 82.15% confirmation accuracy. Adam Optimizer led faster and more stable convergence than SGD.

While regularization improved deep models, some overfit and miscarriage remained, especially between types of similar flowers. The ROC AUC point indicated a room to improve the class separation.

Future work can focus on expanding the dataset, fixing pre -informed layers, discovering new architecture like ResNet and implementing hyperparameter tuning to boost general performance.

9. References

Shorten, C. &. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 28-29.

Sleam, R. (2024, November). *Flowers Dataset*. Retrieved from Kaggle: https://www.kaggle.com/datasets/rahmasleam/flowers-dataset