# Classification of 102 Flowers using Convolutional Neural Networks

Michael Issa

***Abstract*—Convolutional Neural Networks (CNNs) have become indispensable tools in computer vision tasks, particularly in image classification. This paper explores the architecture design and optimization techniques to enhance the performance of a CNN model for flower classification using the Flowers102 dataset. The goal is to develop a model capable of accurately classifying flower images into 102 categories. The study presents five CNN models, each with iterative adjustments in architecture and activation functions, evaluated based on validation and test accuracy scores. Results indicate variations in model performance, with insights into architecture design and optimization strategies for flower classification tasks.**

### *Introduction*

Convolutional Neural Networks (CNNs) have become indispensable tools in computer vision tasks, particularly in image classification. This paper explores the architecture design and optimization techniques to enhance the performance of a CNN model for flower classification using the Flowers102 dataset. The goal is to develop a model capable of accurately classifying flower images into 102 categories.

The task involves classifying images of flowers into 102 categories using CNNs. The Flowers102 dataset, comprising images of varying sizes and qualities, serves as the primary data source. The objective is to design a CNN architecture capable of learning discriminative features and achieving high classification accuracy. The dataset is already split into test, train, and validate. The labels consist of numerical labels for files with various images of a species of flower.

### *Major Challenges and Solutions*

The major challenges in this task include designing an effective CNN architecture, mitigating overfitting, optimizing convergence speed, and ensuring robustness to variations in the input data. Solutions involve iterative experimentation and adjustment of the network architecture, activation functions, and optimization algorithms. An issue I found with the dataset was that the labels were not very informative. It required being able to identify the number labeled flowers visually to figure out what species they were. It would have made out of dataset testing easier to know the species.

### *CNN Models*

There are five CNN models in total with the first four being specified by hand and the last being a pretrained model that is fine-tuned for our model. The first CNN architecture comprises six convolutional layers followed by batch normalization and max-pooling layers. This architecture aims to capture hierarchical features in the input images. The convolutional layers extract low-level features, while the subsequent batch normalization layers help stabilize and accelerate the training process by normalizing the activations. Max-pooling layers downsample the feature maps, reducing computational complexity and aiding in translation invariance. The second introduces the Exponential Linear Unit (ELU) activation function instead of the Rectified Linear Unit (ReLU) in the baseline model. ELU offers smoother gradients for negative values, potentially mitigating the vanishing gradient problem and improving convergence speed. ELU is known to capture more diverse representations of data, leading to better generalization. The third modified architecture replaces the ELU activation function with the Parametric Rectified Linear Unit (PReLU). PReLU introduces learnable parameters that adaptively determine the slope of the activation function, enabling the model to learn more flexible and data-dependent representations. This can be beneficial for complex datasets where the optimal activation function may vary across different regions of the input space. The fourth architecture maintains the modifications introduced in the previous iterations, along with further adjustments. The number of convolutional layers is reduced to four, each followed by batch normalization and max-pooling layers. This simplification aims to reduce model complexity and potential overfitting, while still allowing the network to learn discriminative features effectively. The number of output channels in each convolutional layer is adjusted to optimize feature extraction. The PReLU activation function is retained for its adaptive properties and potential for capturing diverse representations. The last model is a pre-trained Very Deep Convolutional Network (VGGNet). I thought this would be a good test to compare to my own best performing models.

### *Performance Metrics*

Each model trained on the Flowers102 dataset consists of 102 categories of flower images, with varying sizes and backgrounds, divided into training, validation, and test sets. The model's performance is evaluated using classification accuracy on the validation and test sets. I only report the validation accuracy for the final parameter choice.

Iterative experiments involved modifying the CNN architecture, activation functions, and optimization algorithms, resulting in significant improvements in classification accuracy. We analyze convergence behavior, overfitting tendencies, and generalization capabilities of each model iteration in the notebook where the code can be found. It seemed to be the VGG model, which was intended to be the benchmark, overfit quite a bit on the training data. This could be due to the depth

| Model | Validation Accuracy | Test Accuracy |
|---|---|---|
| 1st | 39.41% | 35.68% |
| 2nd | 26.37% | 21.56% |
| 3rd | 32.45% | 28.49% |
| 4th | 31.67% | 25.64% |
| VGG | 36.00% | 5.78% |

TABLE I
VALIDATION AND TEST ACCURACY SCORES

of the network and the lack of a representative dataset. To me, the flowers tended to be quite different in a species and each species had about 20 representative pictures of it. This may not have been enough. Surprisingly our best performing model was the first one with a test accuracy of 35.68%.

The paper demonstrates the importance of architecture design and optimization techniques in enhancing CNN performance for flower classification tasks. Future work may involve exploring more sophisticated architectures, incorporating transfer learning, and investigating novel optimization algorithms to further improve classification accuracy and robustness.