Physical Education, Health and Social Sciences

https://journal-of-social-education.org E-ISSN: 2958-5996

P-ISSN: 2958-5988

Multi-Layer Vision Transformer Approach for Identification and Classification of Flowers

Muhammad Hassan Raza[1],Sajid Ali[2]*, Shahbaz Hassan Wasti

[1] NFC, Institute of Engineering and Technology, Multan, eMAIL: hr5394268@gmail.com,
[2] Department of Information Sciences, University of Education, Multan Campus,
Email: sajid.ali@ue.edu.pk, shahbazwasti@ue.edu.pk
*Corresponding Author: Sajid Ali (sajid.ali@ue.edu.pk)

DOI: https://doi.org/10.63163/jpehss.v3i1.940

Abstract

Beauty is incomplete without flowers. Flowers have been employed for many human-beneficial purposes. Currently, roughly 400,000 different flower kinds are available worldwide. The similarity in shape and color amongst the blooms causes them to vary from one another. The classification of flowers is a challenging subject because of the variety of their shapes, color distribution, illumination, and exposure deformation. Vision Transformer model is applied to explore the flowers shape and color with inter and intra similarity parameter for classification. The two datasets are tested for performance. One is 5-category flowers dataset that is already labeled publicly, and second one is Oxford 102 Flowers. Oxford flower dataset is preprocessed using SIFT algorithm to gain Isomap of different flowers parameters like as color features and shapes. After getting the parameters, the Vision Transformer (ViT) model is applied on feature parameters and our model achieve the competitive and reliable results. In addition, the proposed model has capability to identify and classify the flowers category, also produced necessary details including as flower classification and name for identification. The highest accuracy achieved through proposed model is 97.5% on 5-category flowers dataset and 99.31% on Oxford 102 dataset.

Keywords: Flower Classification, ViT, Vision Transformer, Classification, Agriculture, Plants Identification

Introduction

Understanding and documenting the diverse range of flower species holds immense significance in preservation and managing biodiversity. Consequently, possessing a comprehensive comprehension of flower species becomes imperative for the protection of biodiversity. The global array of countries hosts thousands of flower species, each exhibiting a rich variety. Yet, manually identifying this vast assortment of flowers poses a time-intensive and formidable challenge, even for seasoned botanical experts. A prominent area of study in the domains of computer vision and machine learning revolves around the classification of flowers. This involves the development of methodologies and algorithms designed to autonomously categorize and recognize flowers based on their visual attributes. The practical applications of flower classification span agriculture, horticulture, and biology. This discipline's historical roots trace back to the 18th century when Swedish botanist Carl Linnaeus introduced a taxonomic framework for categorizing plants [1]. Linnaeus's approach was rooted in the physical traits of plants, encompassing attributes like leaf arrangement, floral structure, and fruit characteristics. Over time, flower classification has evolved, embracing more advanced techniques such as computer vision and machine learning. Recent advancements in computer vision technology have propelled the field of flower classification forward. This progress holds particular relevance due

to the resemblance often observed among various flower types in terms of shape, color, and texture. Notably, aquatic plants like lotuses and water lilies exemplify this phenomenon. Numerous methodologies have emerged for image classification, categorically falling into two clusters: conventional machine learning approaches and deep learning techniques. In the first category, raw image can be directly fed into Convolutional Neural Networks (CNNs) with minimal preprocessing, rendering these methods effective in recognition tasks [2]. In contrast, the second group entails transforming raw images into a suitable format, facilitating the extraction of handcrafted features like color, shape, and texture by machines [3]. Moreover, CNNs possess the ability to autonomously learn hierarchical features, serving the objective of image classification or segmentation. In this paper, our focus shifts from recognizing numerous unrelated categories to addressing the challenge of classifying a vast array of classes all falling under the umbrella of flowers. Distinguishing between different types of flowers presents an additional layer of complexity compared to categories like bicycles, automobiles, and felines. This increased difficulty arises due to the significant resemblances shared among flower classes. Adding to this complexity is the fact that flowers are flexible entities capable of undergoing various deformations, leading to substantial intraclass variation. While prior studies on flower classification have primarily tackled a limited number of classes, typically ranging from 10 to 30, our contribution is the introduction of a dataset encompassing 102 classes for flower classification. We propose the utilization of the Vision Transformer technique for the task of classifying and identifying these diverse flower types. We use the self-consideration component to a dataset centered around named images in order to evaluate the convolutional Vision Transformer technique's performance against the most recent CNN-based models. In addition, we present statistics showing that the Vision Transformer method consistently performs better than CNN-based techniques, both on bigger and smaller image datasets.

Literature Review

Recently, researchers have been utilizing the Convolutional Neural Network (CNN) model for a wide array of classification tasks, including but not limited to disease diagnosis, facial and object recognition, as indicated by Voulodimos et al. in 2018 [4]. In the realm of categorizing images of flowers within literary contexts, a variety of strategies and methodologies have emerged. The primary procedures for flower classification involve segmentation, feature extraction, and classification. Several more methods have been proposed for the classification of flowers. These include techniques such as saliency-driven multi-scale nonlinear diffusion filtering, generalized max pooling (GMP) coupled with Fisher Vectors (FV) and power normalization, pairwise rotation invariant co-occurrence local binary pattern (PRICoLBP) [5], metric forests with Gaussian Mixture Models (GMM) [6], visual adjectives (VAs) using Scale-Invariant Feature Transform (SIFT) and improved FV [7], and saliency-driven multi-scale nonlinear diffusion filtering. Further approaches include contextual exemplar classifier (CEC) [10], heterogeneous co-occurrence features [8], generalized hierarchical matching (GHM) in conjunction with saliency map (LocSaliency) [9], and Fisher discrimination dictionary learning (FDDL) in conjunction with frequent local histograms (FLHs). [8], color attention-based bag-of-words [12], Harr-like modification of local features [13], grid-specific bag- of-FLH (GRID-FLH) [11], and graph-regularized robust late fusion (GRLF) [14]. All these techniques use classic classifiers such as Support Vector Machines (SVM) and rely on features that are manually created. The subsequent model resembled the initial one but was fine-tuned to converge on a smaller subset of data it was trained on. And in [11] authors collected high-resolution flower images from Google Photos, pre-processed them, and employed transfer learning using the Tensorflow platform's inception-v3 model. The accuracy of training without transfer learning was 60.2%, whereas flower recognition accuracy reached 88.3%. In the realm of image classification, particularly for flowers, deep neural networks (DNNs) became a central focus of research [15]. This study delved into DNN techniques in detail, training models on five different flower varieties to achieve up to 90% accuracy.

Authors in [16] introduced a deep learning- based approach for flower image recognition, comprising three main stages: image feature extraction, classification, and image preparation. High-level features were extracted from pre-processed images using the Per-VGG16 model. The outcomes demonstrated that this algorithm outperformed certain existing methods, achieving an excellent classification accuracy of 89.1%. In a prior study [17], a task-driven pooling (TDP) model was introduced to implicitly learn pooled representations from data. TDP served as a replacement for the conventional average or max pooling techniques in CNN models, resulting in enhanced pooled representations. This method was further extended to encompass multi-task classification (mTDP) with the primary goal of maximizing accuracy on a flower dataset. Furthermore, an independent study [18] examined how to transfer CNN characteristics to target tasks in an efficient manner; their evaluation showed state-of- the-art performance gains on multiple datasets, including a floral dataset. More recently, a novel strategy described in [19] demonstrated a way to use winner-takes-all (WTA) hashing to speed up the computational efficiency of both the forward and backward propagation stages in CNNs. Conversely, a unique hierarchical deep semantic representation (H- DSR) was proposed in [20], combining semantic context modelling with visual data. Using pre-learned classifiers, this method involved extracting deep CNN features from preset spatial picture grids in order to determine a response map. All things considered; the experiments covered in this paper demonstrate how valuable machine learning algorithms are for classifying flowers according to their visual characteristics. While some studies concentrated on conventional computer vision techniques like color and texture analysis, others made use of deep learning techniques like convolutional neural networks (CNNs) and transfer learning. These methods were evaluated on a variety of datasets, from little flower collections to sizable archives such as ImageNet. The accuracy of these models often matched or surpassed human experts, with outcomes varying based on specific techniques and datasets. Furthermore, some investigations investigated potential applications of flower classification in fields like horticulture, ecology, and conservation.

Proposed Methodology

The emergence of ViT represents a major breakthrough in the realm of computer vision, piloting in fresh possibilities for deep learning models capable of analyzing images as assemblies of tokens. Advancements in machine learning, mirroring the latest progress in computer vision, have managed to attain cutting-edge accuracy levels while optimizing parameter efficiency.

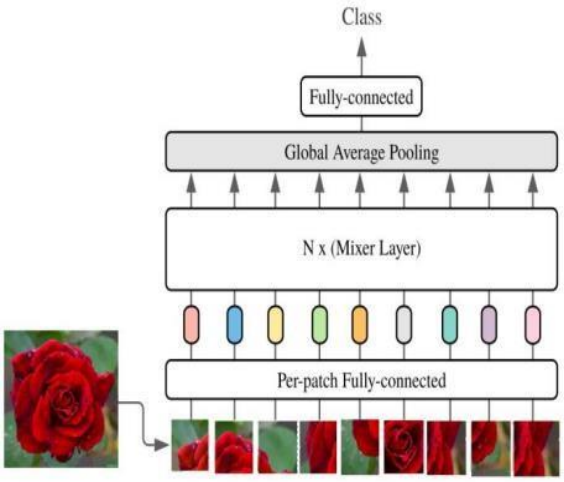

Figure 1: Proposed ViT model diagram of our research work.

This signals a substantial potential for a universal learning approach that can be harnessed across various data formats. ViT enables models to gather and use information on their own. The transformer concept, initially intended for natural language processing, can now be used for computer vision applications through the use of the VIT (Vision Transformer) neural network architecture. Models are divided into size categories that range from "tiny" or "small" to "large" or "extra-large." "Vision Transformer" (short for "Vit") refers to the architecture model. The same self-attention method employed in Transformers for natural language processing is applied to visual input in a more modern class of models known as Vision Transformers. The descriptor "base" pertains to the model's size, indicating that it is one of the smaller options within this architecture. The process of dividing the input image into smaller units is termed "patch16." In this case, the input image is subdivided into 16x16 pixel patches. The dimension "224" specifies the size of the input images, signifying that the model is trained to handle images with dimensions of 224x224 pixels.

Data Description

Our research employs two distinct flower datasets. The first dataset comprises flowers categorized into five distinct classes, while the second dataset is known as the Oxford 102 flowers dataset, encompassing a total of 102 distinct flower classes. Within each of these classes, there are varying quantities of images, with 40 images in some classes and 258 images in others. These images encompass a range of attributes, including scaling, lighting, and positional variations. Notably, some classes exhibit substantial similarities, while others display significant variations. This dataset presents a notably more complex challenge than the 5-category dataset. The complexity arises not only due to the increased number of classes but also because of the reduced number of samples per class and the presence of shared characteristics among several classes.

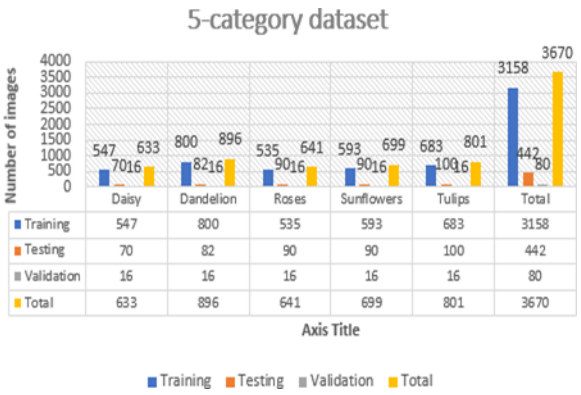

Figure 2: Graphical visualization of 5-category flowers dataset.

To facilitate easy identification of the flower classes, the 102 classes in the dataset are numerically labeled from 1 to 102. This labeling scheme allows for quick recognition of the class to which each flower belongs. The corresponding numerical labels for each class can be found in Table 1, provided below. If we consider a dataset labeled as 'Y' and denote 'C' as the class variable within this dataset, it comprises a total of 102 distinct classes representing various types of flowers and x is representing images in dataset. We can express this in mathematical terms as follows: Classes

$$= C = \{ c_1, c_2 , c_3 , \dots \dots \dots \dots . c_{102}\} \quad \text{eq (1)}$$

$$Y = \{ yc_1 , yc_2 , yc_3 , yc_4 \dots \dots \dots \dots . yc_{102}\} \quad \text{eq (2)}$$

Within these classes, there exists variation in the number of images allocated to each class. While some classes exhibit similar image counts, others display discrepancies in the number of images assigned. Specifically, the class 'Passion flower' contains the highest number of images, while classes such as 'Eustoma,' 'Mexican Aster,' 'Celosia,' 'Moon Orchid,' 'Canterbury Bells,' and 'Primrose' each consist of the lowest count, specifically 40 images per class. The mathematical representation of this disparity in image quantities across different classes is presented below.

$$Y_{c_{ij}} = \sum_{i,j}^{n} x\, c_{ij} \qquad eq\ (3)$$

where n= {1 ,2 ,3 ,4 102} and I and j belong to number of images in each class. Above equation shows that each class in dataset has n number of images.

Data Preprocessing

We determined the appropriate features to employ and devised optimal strategies for their fusion. Eventually, we chose the most suitable attributes in the form of color and shape variations and scrutinized their performance across a dataset comprising 102 categories of flowers. While certain flowers possess distinct colors, shapes, or patterns, for most, their distinctiveness arises from a combination of these factors. Our goal is to establish a visual lexicon for each of these elements. Employing SIFT features as shape descriptors and HSV as color descriptors, we illustrate the categories within this dataset. Our objective is to construct a lexicon that encapsulates a flower's color. While numerous flowers exhibit an array of colors, many possess unique hues. The color of a flower aids in narrowing down potential species, although it may not definitively determine the exact species. For instance, a yellow flower could be either a Daffodil or a Dandelion, but not a Bluebell. Manipulating hue, saturation, value, and transparency offers various options for adjusting colors. The differentiation between flowers can be achieved by considering the shape of individual petals, their arrangement, and the overall form of the flower. While the Daffodil's petals resemble those of the Windflower, the overall shape diverges due to the tubular corolla in the middle of the Daffodil. Changes in perspective and obstructions alter the perceived shape. Describing shape becomes more complex due to a flower's inherent deformations.

Features Data Visualization

We present two visualizations of the dataset: one based on color and the other on shape. To create these visualizations, we initially extract features for each training image and cluster them using k-means clustering. With a set of cluster centers in hand, we generate frequency histograms of word assignments, xf (i), for each image i and feature f. Using the χ^2 distance, we calculate the dissimilarity between frequency histograms for each training image.

$$\mathbf{D_F}\,(\mathbf{C_1}, \mathbf{C_2}) = \sum_{\mathbf{i,j=i+1}}^{\mathbf{n}} \mathbf{Y_f^2}\,(\,\mathbf{x_f}\,(\mathbf{i}), \mathbf{x_f}(\mathbf{j})\,) \qquad \text{eq (4)}$$

Subsequently, we construct an intra-class distance matrix Df by determining the minimum distance between images from each pair of classes, c1 and c2. The images are randomly selected from the group.

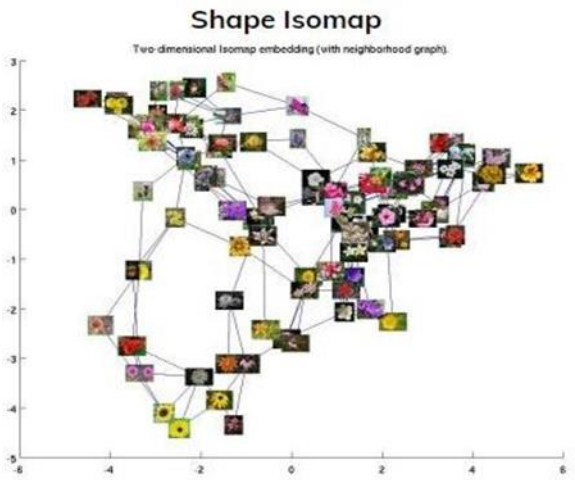

Figure 3: Utilizing Isomap with a K-nearest neighbor approach, we've derived characteristic features with K set to 2. A range of floral compositions, including tubular-shaped flowers with clusters of larger petals, spiky and spherical blooms, and those embellished with countless delicate petals, were revealed in the randomly selected class images that were on display. (sourced from Kaggle).

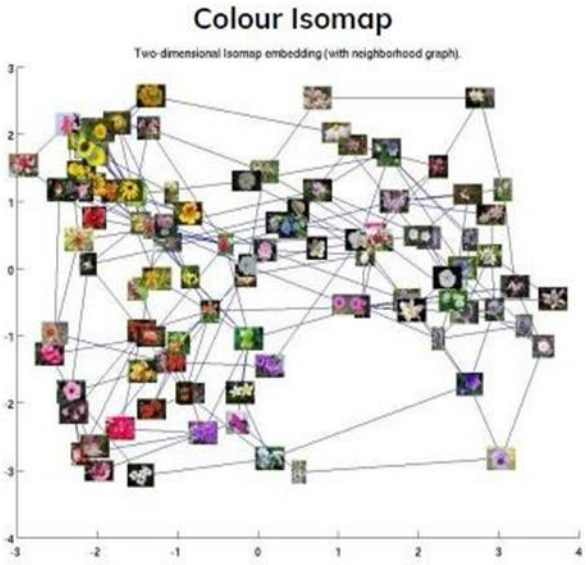

Figure 4: The Isomap figure, incorporating K-nearest- neighbor and color attributes, also utilizes a K value of the displayed class photographs were randomly selected, showcasing a grouping of yellow, pink/red, white, and blue/purple hues (sourced from Kaggle).

Image Input and Patch Generation

For our model to function effectively, it necessitates an input image with dimensions of 224x224 pixels. If the input image's resolution differs from this, the model automatically resizes it to 224x224 pixels. Subsequently, the supplied image is divided into non- overlapping patches, each measuring 16x16 pixels. These patches, generated through this process, serve as the model's input.

Figure 5: 16 Patches grid of image which given to the model as an input.

$$X = \frac{H_X * W_X}{16} \qquad eq\ (5)$$

$$\text{Patch Image (Xp)} \in R_X^{Nx(P_i)} \qquad eq\ (6)$$

$$X = X_1 + X_2 + X_3 \ldots\ldots\ldots\ldots X_N \qquad eq\ (7)$$

$$P_i = P_1 + P_2 + P_3 \ldots\ldots\ldots\ldots P_n \qquad eq\ (8)$$

Firstly, we create a feature vector of shape (n+1, d) by embedding an input image with dimensions (height, width, and channels). The image is divided into n square patches with a shape of (Pi), arranged in a vectorized sequence from top to bottom and left to right. In this case, 'p' stands for a preset value. N = Hx*Wx / Pi is the formula used to calculate the sequence length, N, and the resolution of each image patch, represented as (Rx). Every patch is projected linearly into a D-dimensional space that can be trained. Furthermore, each patch is linearly projected into a vector space to generate embeddings that capture visual characteristics such as color and texture. Following this, positional encoding is applied, and a linear projection is performed on the image patches to facilitate the learning of these embeddings.

$$Y = [X_1P_1L_E;\ X_2P_2L_E;\ \cdots;\ X_NP_N\ L_E] + L_Epo \quad \text{eq (9)}$$

$$L_{E_{POS}} \in R^{(N+1)*D} \qquad eq\ (10)$$

$$D_{X=}\ \ Dx_1 + Dx_2 + Dx_3 \ldots\ldots\ldots\ldots D_{xN} \qquad eq\ (11)$$

In equations 9 and 10, a trainable embedding tensor with dimensions (Pi, D) is employed to linearly project each flattened patch into a d-dimensional space, followed by element-wise multiplication with the flattened patches. This d-dimensionality remains consistent across the majority of components throughout the architecture. The result is an embedded patches with a shape of (1, d). In each of these layers, an attention mechanism is employed to compute a set of weights, determining the contribution of each patch to the overall representation.

$$W = \sum_{i}^{n} (xp_i\ .\ w_i) \qquad \text{eq (12)}$$

In the equation above, 'i' represents the set of 'n' image patches, 'xp' refers to individual image patches, and 'wi' corresponds to the assigned weights for these patches. Based on their placement in the sequence and their position inside the encoding dimension, the positional encoding vectors are calculated. In particular, the positional encoding vector for every position is built by combining sine and cosine functions at different frequencies. As a result, every patch receives a distinct encoding that corresponds to its location within the image.

$$PE_{POS,2i} = \sin \frac{pos}{10000^{2i/d}} \qquad \text{eq (13)}$$

$$PE_{POS,2i} = \cos \frac{pos}{10000^{2i/d}} \qquad \text{eq (14)}$$

In equations 13 and 14, the term "pos" represents the positional embedding for the first word, and "d" signifies the dimension of the word/token embedding, which in this case is set to 5. It's important to note that I denote each of the five distinct dimensions of the embedding. The values of 'pos' and 'I' can vary, while 'd' remains constant. A common neural network architecture used for classification and regression problems is the Multi- Layer Perceptron (MLP). The Vision Transformer (ViT) encoder is a reliable option for image recognition applications; however, it might not always work at its best on its own. Although it is a basic part of many neural network architectures, the Multi-Layer Perceptron (MLP) is not frequently used in the ViT architecture. ViT's feedforward layers handle input data using both linear and non-linear activation functions, like GELU, rather than just linear transformations.

$$X = L_{N(x)} \qquad \text{eq (15)}$$

$$Z = W_2 * G(w_1 * x + b_1) + b_2 \qquad \text{eq (16)}$$

In this context, 'x' represents the input to the layer, 'GELU' stands for the Gaussian Error Linear Unit activation function; 'Layer Normalization' is a function used for normalizing layers, which helps stabilize the training process; 'W1', 'W2', 'b1', and 'b2' are trainable variables.

Experimental Results Analysis

We use evaluation metrics in our experiments to measure our proposed method's performance. Two datasets are used in these experiments: the Oxford 102 classes dataset from Kaggle and a 5-category dataset. On these two datasets, the Vision Transformer has shown better performance than other approaches currently in use. Evaluation metrics like precision, recall, and F1 score are frequently used to assess how well a binary classification model is performing. The numbers from a confusion matrix, which includes True Positives (TP), True Negatives (TN), False Positives (FP), and False

Negatives (FN), are used to build these measures. A perfect score is 1, and lower values denote poor performance. Precision, recall, and the F1 score are all scaled between 0 and 1. Recall measures the model's capacity to predict positive occurrences, while precision measures how well the model finds positive outcomes in the dataset. The F1 score aggregates recall and precision into a single number to provide an overall assessment of the model's performance.

$$\text{Precision} = \frac{T_{positive}}{T_{positive} + F_{positive}} \qquad \text{eq (17)}$$

$$\text{Recall} = \frac{T_{positive}}{T_{positive} + F_{negative}} \qquad \text{eq (18)}$$

$$\text{F1 Score} = 2 \times \frac{\text{Pre} \times \text{Rec}}{\text{Pre} + \text{Rec}} \qquad \text{eq (19)}$$

Table 1: Results of ViT on Datasets

5 Category Dataset		**Oxford 102 Dataset**	
Parameter	Value (%)	Parameter	Value (%)
Training Accuracy	99.90	Training Accuracy	98.79
Testing Accuracy	97.50	Testing Accuracy	99.31
Training Loss	0.03	Training Loss	0.14
Testing Loss	0.10	Testing Loss	0.08

5-category dataset results

For the 5-category dataset, we analyze the performance using a confusion matrix, which is essential to assess the classification models' effectiveness on specific test data. It's important to note that the matrix itself is easy to understand, but some terminology used in relation to it may be less familiar. In certain contexts, it's referred to as an "error matrix" because it presents the model's performance errors in a matrix format. Figure 6 and 7 below illustrates these graphs, showing the validation loss and accuracy for batches.

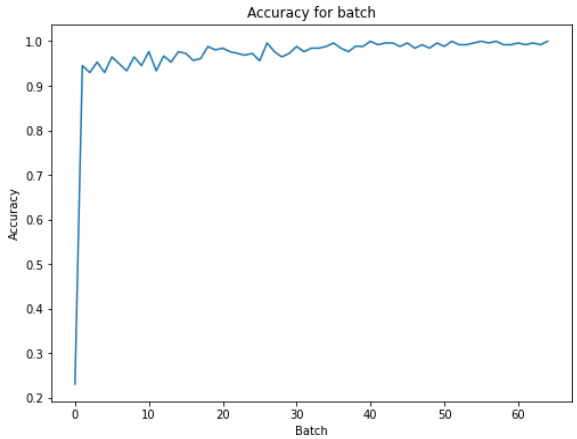

Figure 6: Visualizing Batch Validation Accuracy.

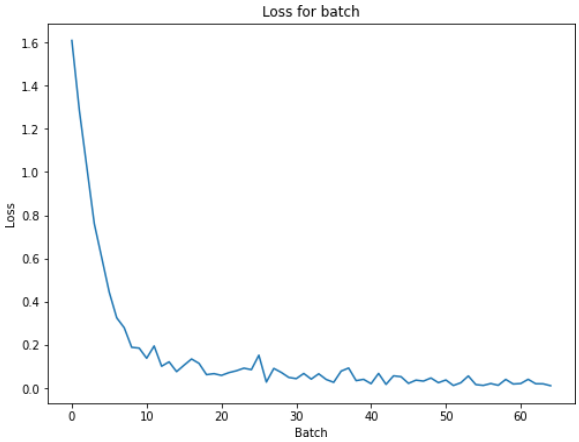

Figure 7: Visualization of Batch Validation loss.

The highest accuracy achieved by our model is 97.50%. Furthermore, another testing method was applied in this study to predict flower classes and determine the accuracy of these predictions. This method involved taking input from the testing directory, selecting random flower samples, and having the model predict their class along with indicating whether the prediction was accurate or not.

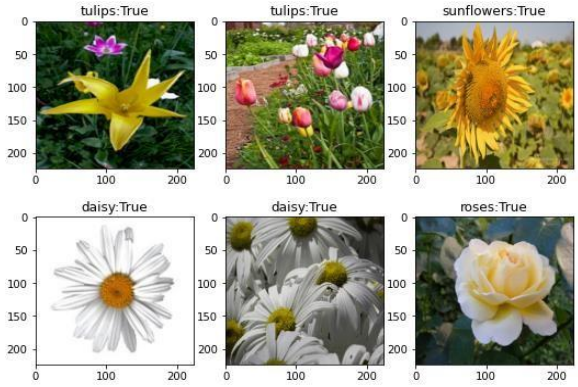

Figure 8: Predicted testing samples of blossoms with the classification labels.

Oxford-102 Flowers Dataset Results

The Oxford 102 category dataset is a widely employed benchmark for image classification challenges, encompassing 102 unique image categories that span a spectrum of complexities and uncertainties. In the realm of image classification, the Vision Transformer model (ViT) stands out as a prominent category of deep learning techniques. Notably, Vision Transformers outperform state-of-the-art approaches and have consistently achieved an impressive average accuracy rate of 99.31% on this dataset. The code section includes graphs depicting validation loss and validation accuracy. Figure 9 and 10 below illustrates these graphs, showing the validation loss and accuracy for batches.

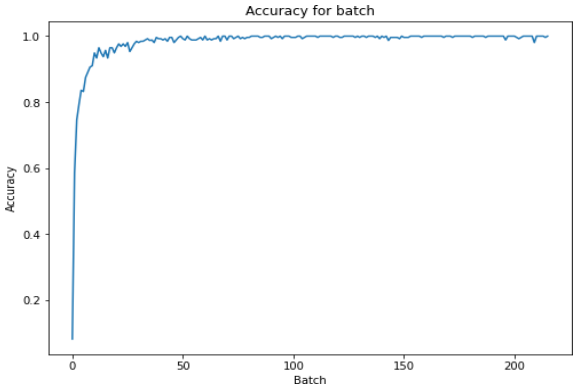

Figure 9: The figure illustrates the accuracy by using the number of epochs that were used in the model. On the Y-axis, our accuracy range is 97–100. The X-axis shows the epochs used to evaluate the model's performance.

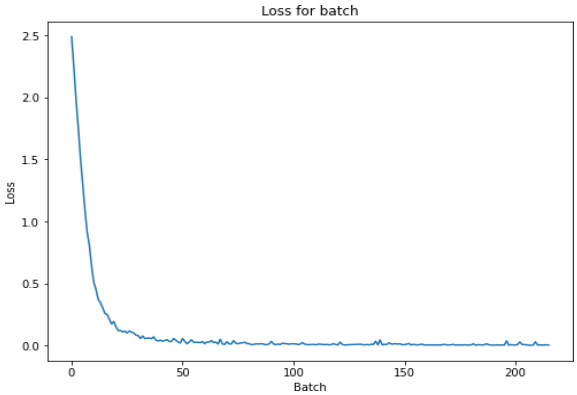

Figure 10: Loss is displayed in the above figure based on the number of epochs that the model ran. The model's performance is shown in epochs on the X-axis.

The results of samples selected at random by the model from the dataset's testing directory are displayed in figure 11 below. These numbers represent the class labels that are provided in the dataset. Additionally, flower class forecasts are shown in figure 12.

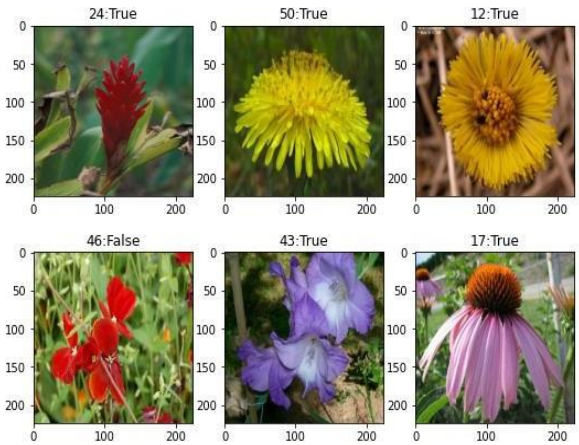

Figure 11: Test dataset predicted samples that indicate the prediction state and label whether the flowers are actually predicted or not.

Table 2: Comparison of Vision Transformer with state-of-the-art techniques

Sr	Dataset	Methods	Accur acy	Referenc e
1	5 category Dataset	DNN	90%	(Abu et al., 2019)
2	5 category Dataset	Inception v2	86%	(Han et al., 2020)h
3	5 category Dataset	Mobile-net v2	96%	(Ball, 2021)
4	5 category Dataset	Yolo V3 & Yolo 5s	95.9%	(Tian & Liao, 2021b)
5	**5 category Dataset**	**Vision Transformer**	**97.5%**	**OURS**
6	Oxford 102	CNN, SVM, VGG16	97.46%	(Gadkari, 2019)
7	Oxford 102	CNN Dense- Net 201	98.36%	(Albardi et al., 2021)
8	Oxford 102	Yolo v5 & Per-VGG16	95.8%	(Tian & Liao, 2021)
9	Oxford 102	Dense-net 121	98.6%	(Desai et al., 2022)
11	**Oxford 102**	**Vision Transformer**	**99.31%**	**OURS**

Conclusion

This paper delves into the realm of flower categorization through the analysis of photographic images. In pursuit of this objective, we have curated extensive flower databases and devised methodologies for the recognition and classification of flowers. We presented two more datasets specifically designed for the categorization of flowers: a 5- category dataset and a large 102-category dataset, both of which were carefully selected to include often encountered floral species. The 102-category dataset presents a significant difficulty because of variances in size and posture, as well as higher interclass similarity. It is noteworthy since it is the only extensive flower database that can be used directly for flower classification. First, we used the 5-category database to test our features and strategy. Then, we moved the best-performing features to the 102-category database. To address the classification task, we employed a novel image classification technique known as the Vision Transformer, resulting in notably high accuracy levels on both datasets. Specifically, we achieved an accuracy rate of 99.31% on the Oxford 102-category.

Reference

[1] Stevens, P. F. (2003). History of Taxonomy. ELS. https://doi.org/10.1038/npg.els.0003093

[2] K.Bae, J.Park,J.Lee,Y.Lee, and C.Lim, "Flower Classification with Modified Multimodal Convolutional Neural Networks", p. 113455, 2020.

[3] R.Shaparia, N. Patel, and Z.Shah, "Flower classification using texture and color features", 2: p. 113-118, 2017

[4] Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep Learning for Computer Vision: A Brief Review. Computational Intelligence and Neuroscience, 2018.

[5] https://doi.org/10.1155/2018/7068349

[6] Shi, L., Li, Z., & Song, D. (2019, February). A flower auto-recognition system based on deep learning. In IOP Conference Series: Earth and Environmental Science (Vol. 234, No. 1, p. 012088). IOP Publishing

[7] Chen, Q., Song, Z, Hua, Y., et al.: 'Hierarchical matching with side information for image classification'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Providence, RI, June 2012, pp. 3426–3433

[8] Qi, X., Xiao, R., Li, C., et al.: 'Pairwise rotation invariant co-occurrence local binary pattern', IEEE Trans. Pattern Anal. Mach. Intell., 2014, 36, (11), pp. 2199–2213

[9] Murray, N., & Perronnin, F. (2014). Generalized max pooling. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2473-2480).

[10] Patel, I., Patel, S. (2020). An Optimized Deep Learning Model for Flower Classification Using NAS-FPN and Faster RCNN. International Journal of Scientific & Technology Research, 9(03), 5308- 5318.

[11] Krizhevsky, A., Sutskever, I., Hinton, G.: 'ImageNet classification with deep convolutional neural networks', in Pereira, F., Burges, C., Bottou, L., et al. (ED.): 'Advances in neural information processing systems' (Curran Associates, Inc., Red Hook, NY, USA, 2012), pp. 1097–1105

[12] Simonyan, K., Zisserman, A.: 'Very deep convolutional networks for largescale image recognition'. Proc. Int. Conf. Learning Representations, San Diego, CA, May 2015, arXiv preprint arXiv:1409.1556

[13] Khan, F. S., Van de Weijer, J., & Vanrell, M. (2012). Modulating shape features by color attention for object recognition. International Journal of Computer Vision, 98(1), 49-64.

[14] Shelhamer, E., Long, J., Darrell, T.: 'Fully convolutional networks for semantic segmentation', IEEE Trans. Pattern Anal. Mach. Intell., 2017, 39, (4), pp. 640–651

[15] Girshick, R., Donahue, J., Darrell, T., et al.: 'Rich feature hierarchies for accurate object detection and semantic segmentation'. Proc. IEEE Conf. Computer Vision and Pattern Recognition, Columbus, OH, June 2014, pp. 580–587

[16] Abu, M. A., Indra, N. H., Rahman, A. H. A., Sapiee, N. A., & Ahmad, I. (2019). A study on image classification based on deep learning and tensorflow. International Journal of Engineering Research and Technology, 12(4), 563–569.

[17] Tian, M., & Liao, Z. (2021a). Research on flower image classification method based on YOLOv5. Journal of Physics: Conference Series,2024(1),3–6.

[18] Xie, G., Zhang, X., Shu, X., et al.: 'Task- driven feature pooling for image classification'. Proc. IEEE Int. Conf. Computer Vision, Santiago, Chile, December 2015, pp. 1179–1187

[19] Zheng, L., Zhao, Y., Wang, S., et al.: 'Good practice in CNN feature transfer', arXiv preprint arXiv:1604.00133, 2016

[20] Bakhtiary, A., Lapedriza, A., Masip, D.: 'Winner takes all hashing for speeding up the training of neural networks in large class problems', Pattern Recognit. Lett., 2017, 93, pp. 38–47

[21] Zhang, C., Li, R., Huang, Q., et al.: 'Hierarchical deep semantic representation for visual categorization', Neurocomputing, 2017, 257, pp. 88– 96