Classification of flower over Oxford 102 dataset

Shristi (12201032019), Akanksha Verma (11301032019) and Sanjana Singh (08901032019)

April 2022

Table 1: Contributions of team members.

Student Name	**Enroll Number**	**Contributions**
Shristi	12201032019	(1) summaries of research papers in related work (2) Wrote Introduction section.(3) Dataset Description and Proposed Methodology (4) Experiment Design, Abstract, Proposed Workflow .
Akanksha Verma	11301032019	(1) Preprocessing of images and Downsampling (2) K Nearest Neighbour(KNN) (3) Transfer Learning (4)Data Augmentation
Sanjana Singh	08901032019	(1) Neural network(NN) (2) Convolutional Neural Network (CNN) (3) CNN with Regularization (4) wrote Future scope and Conclusion

Abstract

Classification of different categories of flowers based on their significant features is our main purpose. Mengxiao Tian et al. [18] proposed the deep learning model to automatically extract the features of flowers with the combination of softmax classifier and achieved 92 % classification accuracy on oxford 17 dataset. One limitation of their proposed method is that it does not have good generalization. In this paper, we investigate to what extent K-nearest neighbor and deep learning approaches can effectively classify the flowers category having some very similar and dissimilar categories of flowers. In **K-nearest neighbour** we achieved 32 % accuracy , in **Neural network** 39.7 % , in **Convolution neural network** 53.9% , in **CNN with regularisation** 58 % and achieved best accuracy of 75.85 % in **transfer learning** approach. We also performed data augmentation and used that augmented data on transfer learning model and achieved 46.07 % classification accuracy. Our transfer learning model outperformed the baseline with an accuracy of 75.85 % .

1 Introduction

India is rich in flora and fauna due to its varied climatic conditions. Various species of flowers are present and it's quite difficult to remember their names and their uses because of the similarities between them. Features that distinguish one flower from another are colour, texture, shape, number of petals or sepals and many more. For example, rose and sunflower both are different in colour, different in shape and in the number of petals or sepals. Flowers are also widely used in medical sciences for example, hibiscus is widely used in Ayurvedic teas which helps in lowering blood pressure, Plumeria is also used in various ailments namely skin diseases, wounds. [1]. We also use flowers for decoration as well as spiritual purposes. There are various other fields as well where flowers play a significant role [13] [4]. [3] have mentioned about the flower classification problem and algorithms. Previous works [16] and [8] have worked on small number of classes 5,17 respectively. Here we use oxford 102 category flower dataset for the flower classification problem.

Some works carried out in this direction. Hiary et al. [6] proposed two step deep learning classifier for flower classification. Firstly, flower segmentation is modelled by the fully convolution network as a binary classifier. Secondly, they build a vigorous CNN classifier to classify dissimilar flowers. They used 3 datasets to test their proposed methodology i.e. the oxford 102 [11], the oxford 17 [10] and Zou-Nagy [19] and the results exceed 97 % on all the datasets. The method fails on the dataset where there is a close similarity in the appearance of dissimilar flowers of different classes, and a significant difference in the appearance of flowers from the same class.
D.S. Guru et al. [5] investigated the effect of texture features for classifying the flowers. The texture features includes color texture moments, gray-level co-occurrence matrix, and Gabor responses. They used a probabilistic neural network as a classifier. They conducted their experiment on their own dataset of 35 classes of flowers, each with 50 samples. They achieved 79 % classification accuracy for the combination of all the features.
Mengxiao Tian et al. [18] proposed the deep learning model to automatically extract the features of flowers with the combination of softmax classifier and achieved 92 % classification accuracy on oxford 17 dataset. There is one limitation of the proposed method that it does not have good generalization.
Alipour et al. [2] proposed robust deep leaning method for the classification of

flowers. They used transfer learning approach by employing DenseNet121 architecture to distinguish the flowers. They also tried fine-tune method to achieve better accuracy and results showed 98.6 % classification accuracy for 50 epocs.

We propose conventional and deep learning approach to solve the flower classification problem. In conventional approach we are using K-Nearest Neighbours(KNN) algorithm and in deep learning approach Neural Network(NN), Convolution Neural Network(CNN) and Transfer Learning. The KNN algorithm classifies the test or unknown data point by finding most common class among the k-nearest data points. No learning and feature extraction happens in KNN algorithm as compared to NN and CNN. In KNN we performed hyperparameter tuning for k's values. We took k's values from 1 to 40 and from that selected the best k value. In NN and CNN algorithm features extraction happens automatically from the training data. In transfer learning we make use of pre-trained model on a new task. And from pre-trained model we mean, model uses the knowledge learned in previous task to predict the new tasks having similar kind of data .In data augmentation, images were augmented by using different methods such as mirrioring ,hortizontal and vertical flip or by many other waysto increase the data and then train and test this data on our transfer learning model . [15].

We performed our experiment on 102oxford dataset from which we selected 23 classes having at atleast 100 images in each class . In **KNN** we achieved 32 % classification accuracy on k's values of 1 and by manhattan distance metric. In **NN** we achieved 39.7 % accuracy by using input size of $100 \times 100 \times 3$ having 100 neurons in input layer and two hidden layers having 80 and 50 neurons respectively with relu activation function. Output layer having 23 neurons with learning rate of 0.0001 with softmax activation function and categorical crossentropy loss. Trained the model on 50 epochs with batch size of 30. The accuracy is less due to large deviation between the similar class and small deviation within the similar class between the images . In **custom CNN** we achieved 53.92 % classification accuracy by using three convolution layers with 3×3 filter size along with three maxpool layers with filter size 2×2 and categorical crossentropy loss. In **CNN with regularisation** we used L2 regularisation with 0.001 learning rate and achieved 58.51 % classification accuracy . In **transfer learning** we used VGG16 model pre-trained on imageNet dataset of 1000 classes with input shape of (70,70,3) and also introduced four fully-connected layer with 120, 100, 80 and 60 neurons each layer with relu as an activation function. The predictive layer have 23 neurons and softmax activation function and achieved 75.85 % classification accuracy. In **Data Augmentation** we augmented our downsampled dataset by hortizonal and vertical fliping , rotation , zooming , featurewise std normalistaion and in many other ways . We used this augmented data on our best model which is transfer learning model and achieved 46.07% classification accuracy .

The **main contribution** of our work.

- Our approach through transfer learning gives 75.85% accuracy which outperforms the baseline score of 72.8% claimed by Nilsback et al. [11].

2 Related Work

Base paper(s): We selected the work of Nilsback et al. [11] as our **base paper** and **Oxford 102 flower dataset**. They investigated to what extent the combination of features can improve the classification performance. They did their computation on four distinct features of flowers and integrated those features using a multiple kernel framework with SVM classifier. Results showed 72.8 % accuracy for the combination of all features as compared to 55.1 % for the single best feature.

Important paper(s): We describe works that have used the **same dataset** as used by the base paper. In Table 2, we compare results of these important papers.

Solanki et al. [17] proposed the method which is divided into two distinct steps for flower classification. Firstly they segmented the flower images and secondly fed those segmented images as an input to CNN model. They used PyTorch library for the purpose of flower recognition. They used various pre-trained models on imageNet dataset and out of which DenseNet achieved 97.92 % highest accuracy.

Liu et al. [9] also worked on the flower classification problem. They used the convolution neural network method for automatic learning of significant features for flower classification. They achieved 76.54 % classification accuracy on their challenging flower dataset and 84.02 % accuracy on oxford 102 flower dataset.

Other papers: In Table 3, we briefly explain prior works who have used different dataset but the same problem.

Nilsback et al. [10] proposed to what extent "bag of visual words" can be used to differentiate the categories having remarkable visual similarity. Their novelty lies in the vocabulary used and how these vocabularies are combined into a final classifier. They have used dataset of 1360 images consisting of 17 flower species.

Liu et al. [8] also worked on effective flower classification problem using SVM classifier. To extract important features of flowers like color, texture and contour information a fused descriptor based on Pyramid Histogram of Visual Words (PHOW) was used. They achieved 86.17 % accuracy on oxford 17 dataset.

Shukkla et al. [16] proposed some machine leaning algorithms for flower classification. On iris dataset they used logistic regression, Neural Network, Support Vector Machine and k-Nearest Neighbor algorithms for the classification.

Kulkarni et al. [7] they proposed the concept of vocabulary so that flower classification task becomes easy . Separate vocabularies for each feature and then combined into final classifier. Classification was done with multiboost classifier.

Sadati at al. [14] proposed convolution neural network (CNN) method for flower classification. They used pre-trained CNN model but removed classification part instead used global average pooling (GAP) in the last layer for feature extraction. Concatenated features from those models then used by SVM for flower classification. They used both oxford 102 and 17 flower dataset and achieved 96.47 % accuracy on oxford 102 flower and 97.64 % accuracy on oxford 17 flower datset.

Peryanto et al. [12] proposed both Convolution Neural Network (CNN) and Support Vector Machine (SVM) method for flower classification. CNN performed well over SVM with an accuracy of 91.6 %, precision of 91.6, recall of 91.6 and F1 Score of 91.6. Images used for classification were taken from google.

Table 2: Comparative analysis between prior research works who have used **same dataset** as the basepaper.

Author(s)	**Brief Description**	**Results**
Solanki et al. [17] [a]	proposed the method which is divided into two distinct steps for flower classification. Firstly they segmented the flower images and secondly fed those segmented images as an input to CNN model.They used PyTorch library for the purpose of flower recognition. They used various pre-trained models on imageNet dataset	out of many models DenseNet achieved 97.92 % highest accuracy.
Liu et al. [9] [b]	worked on the flower classification problem. They used the convolution neural network method for automatic learning of significant features for flower classification.	Achieved 84.02 % accuracy on oxford 102 flower dataset.

[a] https://link.springer.com/chapter/10.1007/978-981-19-0284-0$_1$7
[b] https://ieeexplore.ieee.org/abstract/document/7818296

Table 3: Prior research works related to our problem but using different datasets

Author(s)	**DataSet**	**Brief Description**
Nilsback et al. [10]	Used dataset of 1360 images consisting of 17 flower species. [1]	They proposed to what extent "bag of visual words" can be used to differentiate the categories having remarkable visual similarity. Their novelty lies in the vocabulary used and how these vocabularies are combined into a final classifier.

[1] https://www.robots.ox.ac.uk/ vgg/data/flowers/17/

Liu et al. [8]	Used Oxford 17 dataset . [2]	Worked on effective flower classification problem using SVM classifier. To extract important features of flowers like color, texture and contour information a fused descriptor based on Pyramid Histogram of Visual Words (PHOW) was used. They achieved 86.17 % accuracy.
Shukkla et al. [16]	Used Iris dataset [3]	Proposed some machine leaning algorithms for flower classification. On iris dataset they used Neural Network, Logistic Regression, Support Vector Machine and k-Nearest Neighbors algorithms for the classification.
Kulkarni et al. [7]	Used Oxford 17 dataset.[4]	They proposed the concept of vocabulary so that flower classification task becomes easy. Separate vocabularies for each feature and then combined into final classifier. Classification was done with multiboost classifier.

[2] https://www.robots.ox.ac.uk/ vgg/data/flowers/17/
[3] https://archive.ics.uci.edu/ml/datasets/iris
[4] https://www.robots.ox.ac.uk/ vgg/data/flowers/17/

Sadati et al. [14]	Used Oxford 102 and 17 dataset [5] [6]	Proposed convolution neural network (CNN) method for flower classification. They used pre-trained CNN model but removed classification part instead used global average pooling (GAP) in the last layer for feature extraction. Concatenated features from those models then used by SVM for flower classification. They used both oxford 102 and 17 flower dataset and achieved 96.47 % accuracy on oxford 102 flower and 97.64 % accuracy on oxford 17 flower datset.
Peryanto et al. [12]	Used images from google for the dataset	proposed both Convolution Neural Network (CNN) and Support Vector Machine (SVM) method for flower classification. CNN performed well over SVM with an accuracy of 91.6 %, precision of 91.6, recall of 91.6 and F1 Score of 91.6. Images used for classification were taken from google.

3 Dataset Description

We used oxford 102 category flower images dataset [7]. The images were collected by searching the web and taking pictures. There are minimum 40 images for each class.

Table 4 is an example describing the dataset through counts of some key entities involved in the dataset.

[5] https://www.robots.ox.ac.uk/ vgg/data/flowers/102/
[6] https://www.robots.ox.ac.uk/ vgg/data/flowers/17/
[7] https://www.robots.ox.ac.uk/ vgg/data/flowers/102/

Table 4: Dataset description

Details	**Count**
Number of classes	102
Number of images in the classes	40 - 258

Table 5 describes the attributes of the dataset.

Table 5: Details of Data Attributes with example data instance taken from the dataset.

Data Attributes	**Brief Explanation**	**Example**
Class Label	Unique identifier of each class	21
Image description	All images are RGB images with different resolution and belongs to different category. Each category contains minimum of 40 images	Category lotus contains 137 images.

Figure 1: Morning Glory

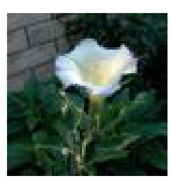

Figure 2: Thorn Apple

4 Proposed Methodology

We have selected the oxford 102 category flower dataset where each category contains images between 40 and 258. There are categories that have significant variations within the classes and several with very similar categories. Out of 102 categories of flowers we have selected 23 categories for our experiment purpose.

4.1 Data Pre-processing

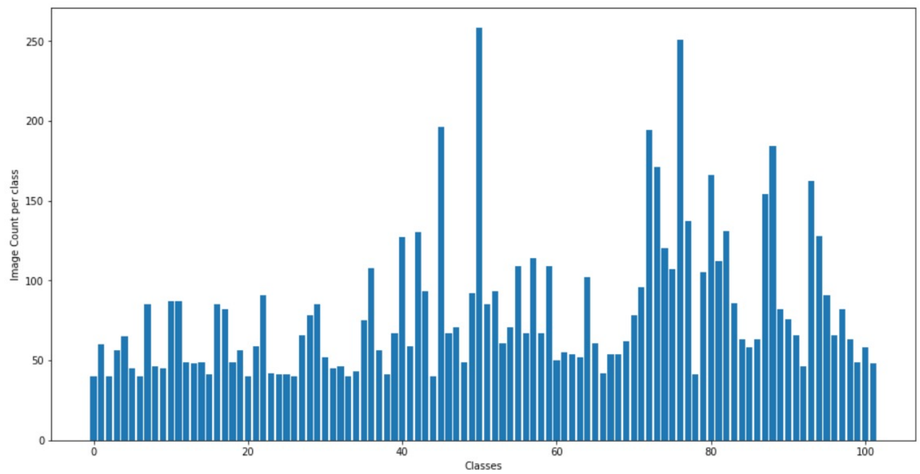

Figure 3: shows the variation in number of images for each class

Oxford 102 dataset contains 102 categories of flowers having images between 40 and 258. It is visible from the histogram that many classes contain images less than 100 which is not sufficient for training the model therefore we have conducted our experiment on 23 classes containing at least 100 images. Also, the dataset contains images with different resolutions therefore to bring the uniformity we re-sized the images to 100×100.

4.2 Proposed Workflow / Architecture

Fig 4 describes the architecture or work flow of our proposed approach. We proposed transfer learning model for flower classification problem. In transfer learning approach we make use of pre-trained model on a new task. We selected VGG16 model, already trained on imageNet dataset. We re-sized the images to 70×70. Provided input shape of (70,70,3). In VGG16 model we added four fully-connected layers with 120 neurons in first layer, 100 neurons in second layer, 80 neurons in third layer and 60 neurons in fourth layer with relu activation function in each layer. Predictive layer consists of 23 neurons with softmax activation function and categorical crossentropy loss. We defined learning rate of 0.0001 as a hyperparameter with adam optimizer during compilation time. We fitted the model on 50 epochs with 84 steps per epoch. We also provided early stopping while monitoring validation accuracy with patience of 5 at max mode.

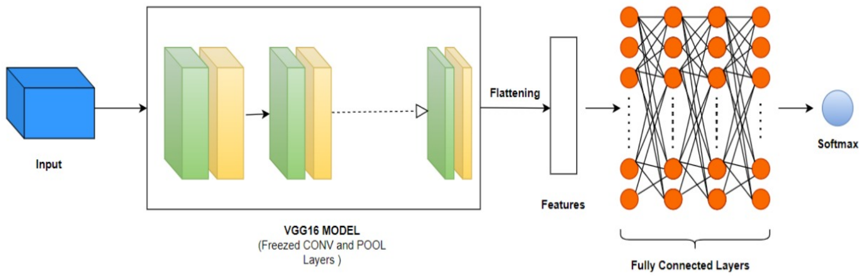

Figure 4: Input given to the model were RGB images which were fed into VGG16 CNN model, pre-trained on imageNet dataset whose convolution and pool layers were freezed and extracting significant features of an image, then those features were fed into fully-connected layers after flattening.

As shown in Fig 4 we passed the images of shape (70,70,3) as an input to our defined model. Firstly the input was given to VGG16 model's convolution and pool layers to extract the significant features of images and then those features were passed into fully connected layers in flattened format where they learnt more weights. After that softmax function predict the probability for each class.
Equation eq(1) shows the softmax activation function used to predict the category of flowers in output layer.

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}} \tag{1}$$

x defines the input vector
k are the number of classes and in our case k's value is 23
x_i defines the i^{th} input
e^{x_i} defines the exponential function for input vector
e^{x_j} defines the exponential function for output vector

5 Experiment Design

We performed our experiment on oxford 102 flower dataset. Fig 5 shows the work flow of experiments performed by using different approaches.

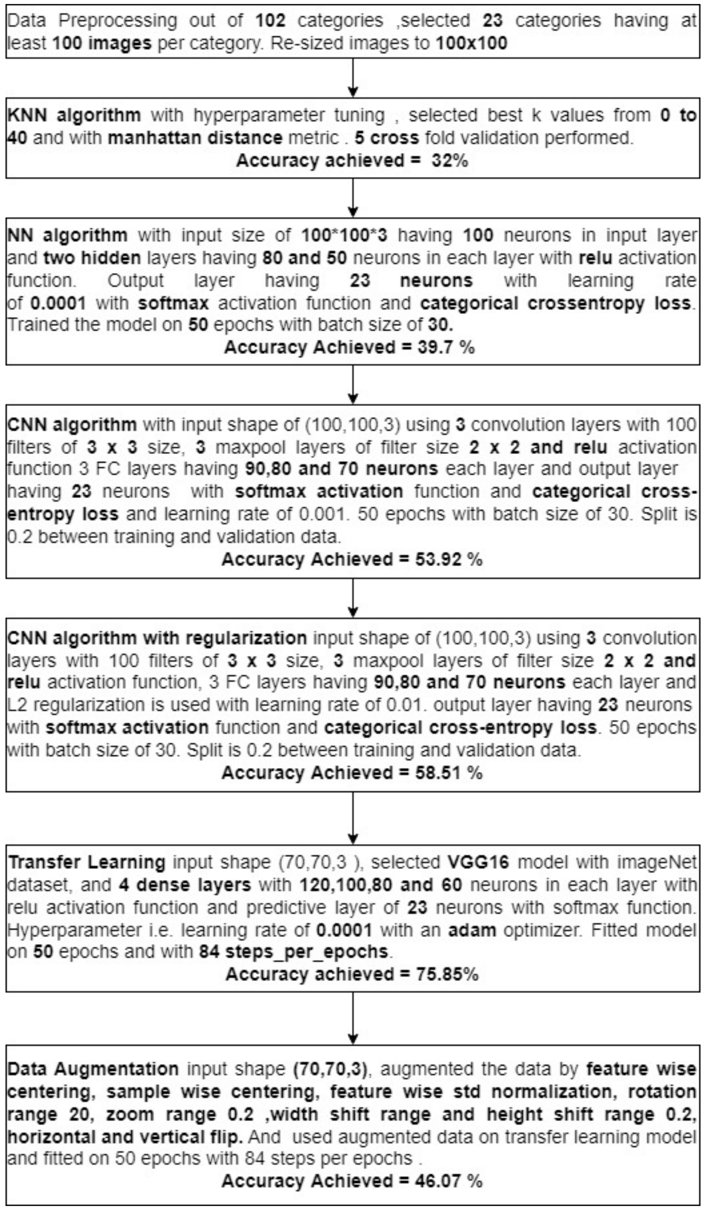

Figure 5: shows the flow diagram of different experiments performed

6 Results & Observations

We selected Nilsback et al. [11] as our **base paper** and **Oxford 102 category flower dataset** to perform our experiments. They achieved 72.8 % accuracy for the combination of features using SVM classifier.

Table 6: Results

Model	**Training Accuracy %**	**Validation Accuracy %**	**Testing Accuracy %**
KNN	100	24.33	32
NN	94.12	42.22	39.7
CNN	100	49.63	53.92
CNN with Regularization	100	62.22	58.51
Transfer learning	100	74.44	75.85
Data Augmentation on Transfer Learning model	80.28	83.93	46.07

We selected oxford 102 category flower dataset and out of 102 we performed our experiment on 23 classes containing minimum of 100 images, as data was not uniformly distributed. We split our dataset into 80:20 ratio in train and test data respectively. We performed data pre-processing on our data i.e resized our image data to $100 \times 100 \times 3$ and in model we passed image data in flatten format i.e (,30000) .
In **knn**, we performed hyperparameter tuning for k's values. We took k's values from 1 to 40 and selected best value of K by plotting error rate vs k values graph and then accordingly selected k value and achieved 32 % classification accuracy on 5 fold cross validation. F1 score , recall and precision are shown below in classification report.

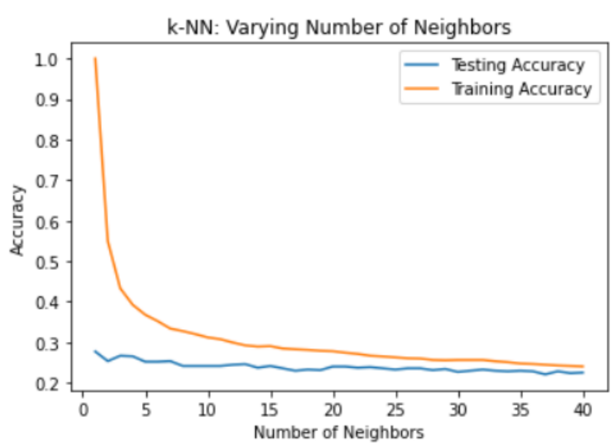

Figure 6: shows the training and validation accuracy of the trained KNN model

Table 7: KNN classification Report

Class	**Precision**	**Recall**	**F1-Score**	**Support**
class 0	0.25	0.73	0.37	22
class 1	0.38	0.24	0.29	25
class 2	0.18	0.15	0.17	26
class 3	0.47	0.18	0.26	39
class 4	0.43	0.17	0.25	52
class 5	0.48	0.91	0.62	22
class 6	0.54	0.87	0.67	23
class 7	0.47	0.36	0.41	22
class 8	0.22	1.00	0.36	20
class 9	0.39	0.23	0.29	39
class 10	0.22	0.41	0.28	34
class 11	0.32	0.75	0.44	24
class 12	0.55	0.29	0.37	21
class 13	0.57	0.08	0.14	50
class 14	0.21	0.67	0.32	27
class 15	0.09	0.05	0.06	21
class 16	1.00	0.09	0.17	33
class 17	0.14	0.04	0.07	23
class 18	0.20	0.12	0.15	26
class 19	0.32	0.19	0.24	31
class 20	0.47	0.38	0.42	37
class 21	0.71	0.16	0.26	32
class 22	0.36	0.15	0.22	26
accuracy			0.32	675
macro avg	0.39	0.36	0.30	675
weighted avg	0.41	0.32	0.28	675

In **Neural Network**, a set of feature is given as the input to the model followed by several hidden layers and activation function is used for introducing non linearity. Our model has one input layer with input size of $100 \times 100 \times 3$ having 100 neurons and two hidden layers having 80 and 50 neurons in each layer with relu activation function and predictive layer have 23 neurons with softmax activation function . We compiled our model with learning rate of 0.0001 and categorical crossentropy loss. We trained the model on 50 epochs with batch size of 30 and achieved 39.7 % classification accuracy.

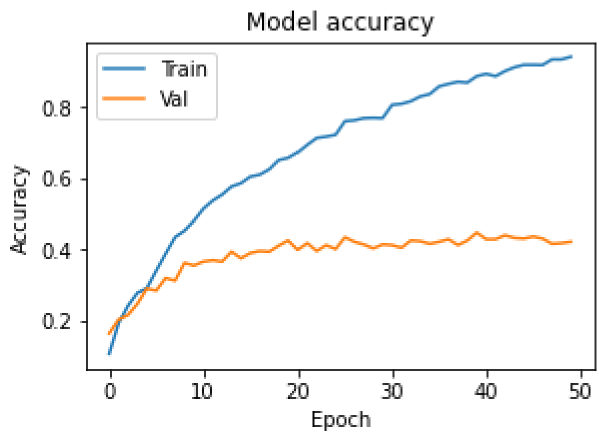

Figure 7: shows the training and validation accuracy of the trained neural network model

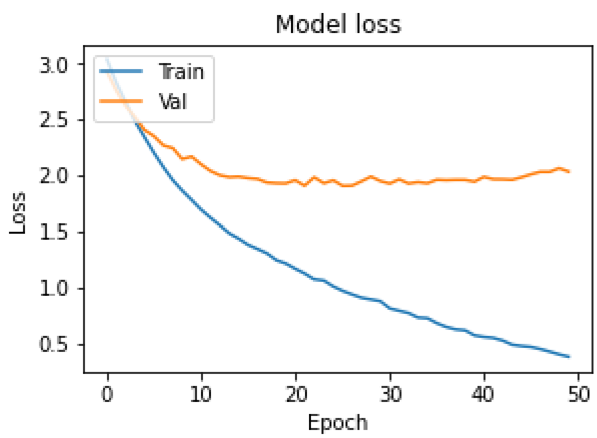

Figure 8: shows the training and validation loss of the trained neural network model.

Table 8: NN classification Report

Class	**Precision**	**Recall**	**F1-Score**	**Support**
class 0	0.35	0.41	0.38	22
class 1	0.35	0.32	0.33	25

class 2	0.24	0.42	0.31	26
class 3	0.50	0.41	0.45	39
class 4	0.35	0.27	0.30	52
class 5	0.71	0.91	0.80	22
class 6	0.50	0.70	0.58	23
class 7	0.35	0.36	0.36	22
class 8	0.44	0.40	0.42	20
class 9	0.35	0.33	0.34	39
class 10	0.30	0.21	0.25	34
class 11	0.73	0.67	0.70	24
class 12	0.59	0.48	0.53	21
class 13	0.66	0.50	0.57	50
class 14	0.32	0.44	0.37	27
class 15	0.22	0.29	0.25	21
class 16	0.47	0.27	0.35	33
class 17	0.30	0.26	0.28	23
class 18	0.21	0.15	0.18	26
class 19	0.29	0.52	0.37	31
class 20	0.43	0.57	0.49	37
class 21	0.32	0.22	0.26	32
class 22	0.29	0.23	0.26	26
accuracy			0.40	675
macro avg	0.40	0.41	0.40	675
weighted avg	0.41	0.40	0.39	675

In **custom cnn model** the input data is (100,100,3,) and contains three convolution layers with 100 filters of 3×3 size, and three pooling layers with 100 filters of size 2×2 followed by three fully connected layers having 90,80 and 70 neurons. The data is split into 80 percent and 20 percent that is training data and validation data respectively. The learning rate of 0.0001 is taken into consideration and relu activation function is being used for introducing non linearity. Output layer having 23 neurons with softmax activation function and categorical cross-entropy loss. It gives 53.92 % accuracy in the flower classification on 50 epochs and batch size of 30.

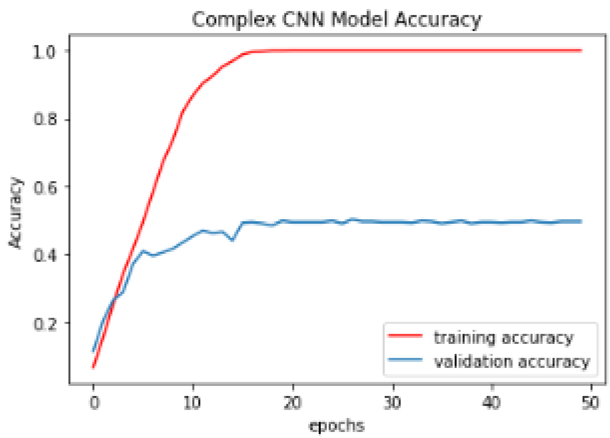

Figure 9: shows the training and validation accuracy of the trained CNN model

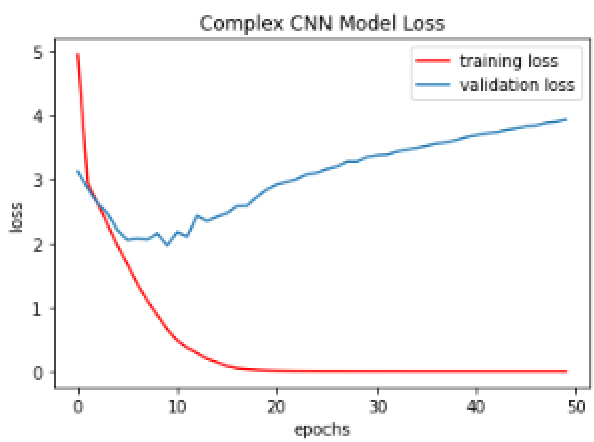

Figure 10: shows the training and validation loss on the trained CNN model

Table 9: Convolution Neural Network classification Report

Class	**Precision**	**Recall**	**F1-Score**	**Support**
class 0	0.55	0.82	0.65	22
class 1	0.38	0.32	0.35	25
class 2	0.34	0.46	0.39	26
class 3	0.64	0.46	0.54	39
class 4	0.57	0.50	0.53	52
class 5	0.90	0.82	0.86	22

class 6	0.78	0.78	0.78	23
class 7	0.74	0.77	0.76	22
class 8	0.46	0.65	0.54	20
class 9	0.58	0.49	0.53	39
class 10	0.42	0.59	0.49	34
class 11	0.56	0.62	0.59	24
class 12	0.61	0.52	0.56	21
class 13	0.78	0.80	0.79	50
class 14	0.66	0.70	0.68	27
class 15	0.27	0.19	0.22	21
class 16	0.65	0.67	0.66	33
class 17	0.58	0.30	0.40	23
class 18	0.31	0.31	0.31	26
class 19	0.61	0.55	0.58	31
class 20	0.52	0.65	0.58	37
class 21	0.44	0.47	0.45	32
class 22	0.53	0.35	0.52	26
accuracy			0.54	675
macro avg	0.53	0.55	0.54	675
weighted avg	0.53	0.54	0.53	675

In **CNN with regularization**, input shape of (100,100,3) using three convolution layers with 100 filters of 3×3 size, threre maxpool layers of filter size 2×2 and relu activation function followed by three fully connected layers having 90,80 and 70 neurons in each layer and L2 regularization is used with learning rate of 0.01. output layer having 23 neurons with softmax activation function and categorical cross-entropy loss. Split is 0.2 between training and validation data. The accuracy achieved is 58.51 % on 50 epochs with batch size of 30.

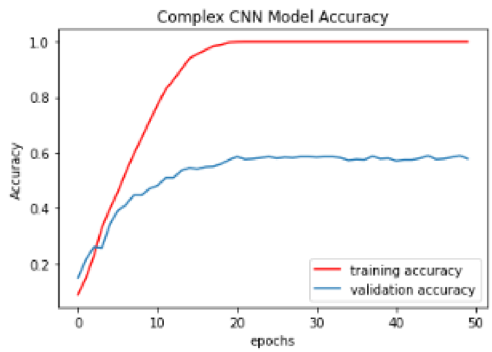

Figure 11: shows the training and validation accuracy on the trained CNN model with regularization

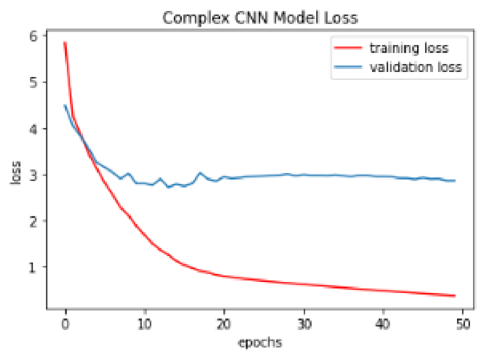

Figure 12: shows the training and validation loss on the trained CNN model with regularization

Table 10: classification Report Convolution Neural Network with Regularization

Class	**Precision**	**Recall**	**F1-Score**	**Support**
class 0	0.65	0.68	0.67	22
class 1	0.50	0.52	0.51	25
class 2	0.31	0.31	0.31	26
class 3	0.62	0.67	0.64	39
class 4	0.45	0.48	0.46	52
class 5	0.79	0.86	0.83	22
class 6	0.81	0.74	0.77	23

class 7	0.68	0.77	0.72	22
class 8	0.64	0.45	0.53	20
class 9	0.68	0.64	0.66	39
class 10	0.51	0.62	0.56	34
class 11	0.63	0.71	0.67	24
class 12	0.40	0.29	0.33	21
class 13	0.85	0.90	0.87	50
class 14	0.63	0.70	0.67	27
class 15	0.30	0.14	0.19	21
class 16	0.67	0.79	0.72	33
class 17	0.48	0.57	0.52	23
class 18	0.26	0.23	0.24	26
class 19	0.70	0.45	0.55	31
class 20	0.60	0.73	0.66	37
class 21	0.55	0.50	0.52	32
class 22	0.50	0.42	0.46	26
accuracy			0.59	675
macro avg	0.57	0.57	0.57	675
weighted avg	0.58	0.59	0.58	675

Transfer learning is a very powerful method where we use pre-trained model on new-task. It makes use of knowledge learned from already trained model on new tasks by increasing the prediction of new tasks i.e decreases the training time . It is a CPU intensive tasks as it requires high CPU power. For transfer learning we used VGG16 model pre-trained on ImageNet dataset, by freezing the convolution and pooling layer followed by four fully connected layers having 120,100,80 and 60 neurons with relu activation function in each layer and a predictive layer with 23 neurons with softmax activation function. We provide learning rate 0.0001 with Adam optimizer and loss as categorical crossentropy . Fitted the model on 50 epochs with 84 steps per epoch and achieved 75.85 % classification accuracy.

Table 11: Transfer Learning Classification report

Class	**Precision**	**Recall**	**F1-Score**	**Support**
class 0	0.23	0.82	0.35	22

class 1	0.96	0.90	0.94	25
class 2	0.67	0.38	0.49	26
class 3	0.92	0.90	0.91	39
class 4	0.70	0.83	0.76	52
class 5	0.75	0.95	0.84	22
class 6	0.96	0.96	0.96	23
class 7	0.96	1.00	0.98	22
class 8	1.00	0.85	0.92	20
class 9	0.83	0.77	0.80	39
class 10	0.74	0.50	0.60	34
class 11	0.75	0.50	0.60	24
class 12	0.81	0.81	0.81	21
class 13	0.90	0.90	0.90	50
class 14	0.68	0.63	0.65	27
class 15	0.74	0.67	0.70	21
class 16	0.87	0.82	0.84	33
class 17	0.60	0.39	0.47	23
class 18	0.75	0.46	0.57	26
class 19	0.70	0.68	0.69	31
class 20	0.81	0.78	0.79	37
class 21	0.83	0.78	0.81	32
class 22	0.50	0.35	0.41	26
accuracy			0.73	675
macro avg	0.77	0.72	0.73	675
weighted avg	0.77	0.73	0.74	675

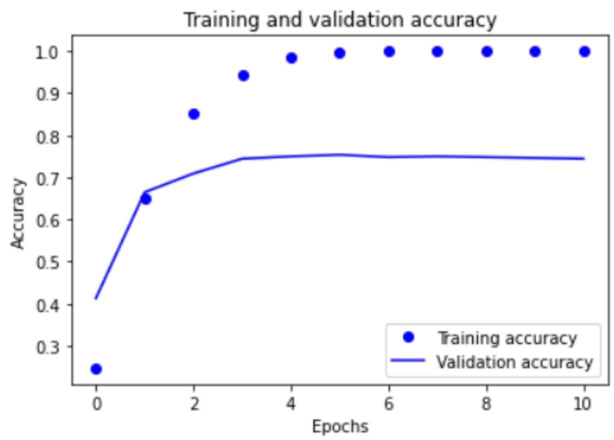

Figure 13: shows the training and validation accuracy of the trained model.

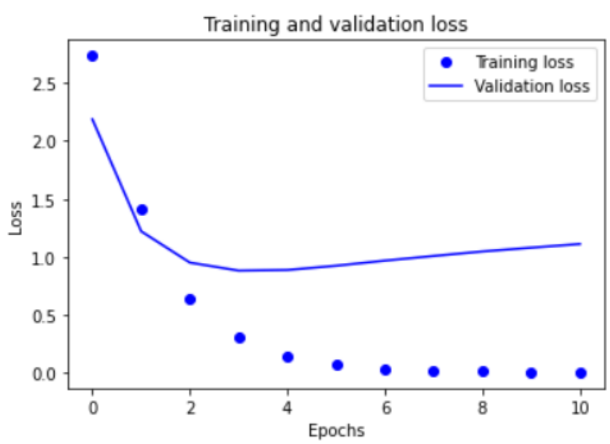

Figure 14: shows the training and validation loss of the trained model.

Data Augmentation is a method to augment data by cropping , mirroring , flipping and by many other ways. We augmented our data by feature wise centering, sample wise centering, feature wise std normalization, rotation range 20, zoom range 0.2, width shift range and height shift range 0.2, horizontal and vertical flip. We used this augmented data on transfer learning model. The model was compiled with learning rate of 0.0001 on Adam optimizer with catergorical crossentropy loss. Model was fitted on 50 epochs with 84 steps per epochs and achieved 46.07 % classification accuracy.

Table 12: Transfer Learning with Data Augmentation Classification Report

Class	Precision	Recall	F1-Score	Support
class 0	0.06	0.95	0.12	22
class 1	0.94	0.64	0.76	25
class 2	0.83	0.19	0.31	26
class 3	0.80	0.10	0.18	0.39
class 4	0.60	0.35	0.44	52
class 5	1.00	0.14	0.24	22
class 6	0.00	0.00	0.00	23
class 7	0.00	0.00	0.00	22
class 8	0.38	0.90	0.54	20
class 9	0.92	0.28	0.43	39
class 10	0.67	0.29	0.41	34
class 11	0.30	0.54	0.38	24
class 12	0.81	0.62	0.70	21
class 13	0.86	0.64	0.74	50
class 14	0.50	0.04	0.07	27
class 15	0.58	0.86	0.69	21
class 16	0.96	0.67	0.79	33
class 17	0.00	0.00	0.00	23
class 18	0.00	0.00	0.00	26
class 19	0.71	0.16	0.26	31
class 20	1.00	0.08	0.15	37
class 21	0.82	0.56	0.67	32
class 22	0.26	0.19	0.22	26
accuracy			0.35	675
macro avg	0.57	0.36	0.35	675
weighted avg	0.61	0.35	0.37	675

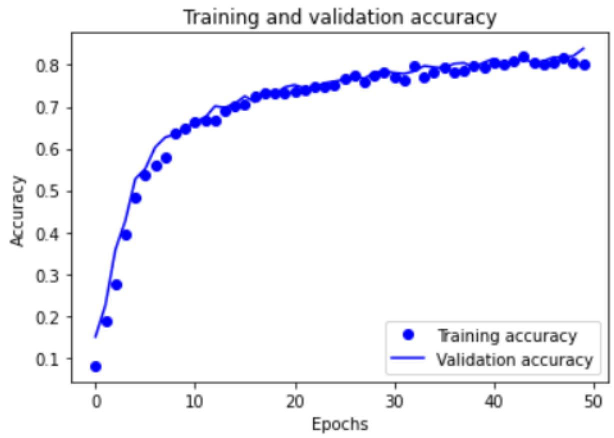

Figure 15: shows the training and validation accuracy of the transfer learning model with data augmentation.

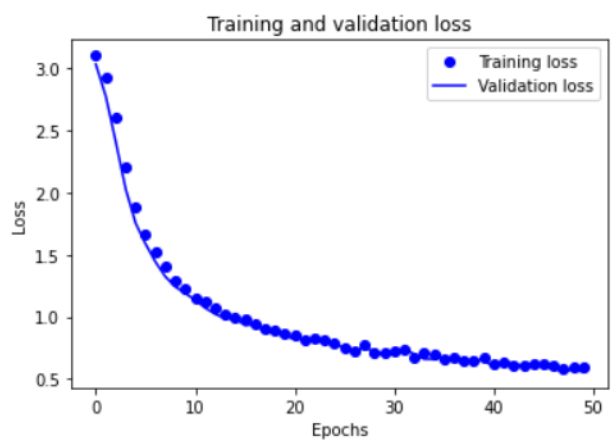

Figure 16: shows the training and validation loss of the transfer learning model with data augmentation.

7 Future Work and Conclusion

In this thesis, we covered the problem of classification of flower in Oxford 102 Dataset using conventional and deep learning approach. We aimed at optimising and improving the classification problem amongst the different flower categories. On the basis of the experiments performed, we propose transfer learning using VGG16 model trained on ImageNet Dataset as the best methodology for classification of flower as we achieved testing accuracy of 75.85 % when the model was fitted on 50 epochs which surpass the baseline accuracy of 72.8 %.

We want to further improvise our classification model using various deep learning approaches. The algorithm having the best efficiency can be integrated with the mobile application for recognition of flower.

Our work in the field of flower classification is of great importance for scientists and botanists who study on enormous variety of flower and their indistinguishable species.

References

[1] Medicinal importance of flowers. "https://bit.ly/3s5LXwV".

[2] Neda Alipour, Omid Tarkhaneh, Mohammad Awrangjeb, and Hongda Tian. Flower image classification using deep convolutional neural network. In *2021 7th International Conference on Web Research (ICWR)*, pages 1–4, 2021.

[3] Prof. Nilesh B. Prajapati Chaku Gamit, Prof. Prashant B. Swadas. Literature review on flower classification. "https://bit.ly/3OMgZ6J", 2015.

[4] Isaiah David. Uses of flowers. "https://bit.ly/3LvFUcx", 2017.

[5] D.S. Guru, Y.H. Sharath Kumar, and S. Manjunath. Textural features in flower classification. *Mathematical and Computer Modelling*, 54(3):1030–1036, 2011. Mathematical and Computer Modeling in agriculture (CCTA 2010).

[6] Hazem Hiary, Heba Saadeh, Maha Saadeh, and Mohammad Yaqub. Flower classification using deep convolutional neural networks. *IET Computer Vision*, 12(6):855–862, 2018.

[7] Tanvi Kulkarni and Nilesh J Uke. Implementation of image based flower classification system. *Int J Comput Sci Bus Inform (IJCSBI)*, 13(1):35–44, 2014.

[8] Wei Liu, Yunbo Rao, Baijiang Fan, Jiali Song, and Qifei Wang. Flower classification using fusion descriptor and svm. In *2017 International Smart Cities Conference (ISC2)*, pages 1–4, 2017.

[9] Yuanyuan Liu, Fan Tang, Dengwen Zhou, Yiping Meng, and Weiming Dong. Flower classification via convolutional neural network. In *2016 IEEE International Conference on Functional-Structural Plant Growth Modeling, Simulation, Visualization and Applications (FSPMA)*, pages 110–116, 2016.

[10] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1447–1454, 2006.

[11] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*, pages 722–729, 2008.

[12] Ari Peryanto, Anton Yudhana, and Rusydi Umar. Convolutional neural network and support vector machine in classification of flower images. *Khazanah Informatika: Jurnal Ilmu Komputer dan Informatika*, 8(1), 2022.

[13] Ava Rose. Unexpected uses for flowers. "https://bit.ly/3F9Uioh", 2019.

[14] Faeze Sadati and Behrooz Rezaie. An improved image classification based in feature extraction from convolutional neural network: Application to flower classification. In *2021 12th International Conference on Information and Knowledge Technology (IKT)*, pages 35–40, 2021.

[15] Pranshu Sharma. Understanding of transfer learning. `"https://bit.ly/3kD6l41"`, 2021.

[16] Asmita Shukla, Ankita Agarwal, Hemlata Pant, and Priyanka Mishra. Flower classification using supervised learning. *vol*, 9:757–762, 2020.

[17] Arun Solanki and Tarana Singh. Flower species detection system using deep convolutional neural networks. In *Emerging Technologies for Computing, Communication and Smart Cities*, pages 217–231. Springer, 2022.

[18] Mengxiao Tian, Hong Chen, and Qing Wang. Flower identification based on deep learning. *Journal of Physics: Conference Series*, 1237(2):022060, jun 2019.

[19] Jie Zou and G. Nagy. Evaluation of model-based interactive flower recognition. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 2, pages 311–314 Vol.2, 2004.