International Journal of Engineering & Technology, *7 (1.1) (2018) 384-391*

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

An efficient classification of flower images with convolutional neural networks

M. V. D. Prasad [1] *, B. Jwala Lakshmamma [1], A. Hari Chandana [1], K. Komali [1], M.V.N. Manoja [1], P. Rajesh Kumar [2], Ch. Raghava Prasad [1], Syed Inthiyaz [1], P. Sasi Kiran [3]

[1] Department of ECE, KLEF-Deemed to be University, Guntur, A.P., India
[2] Department of ECE, AU College of Engineering, Andhra University, Visakhapatnam, A.P., India
[3] Department of EEE, Raghu College of Engineering, Visakhapatnam, A.P., India.
**Corresponding author E-mail: mvd_ece@kluniversity.in*

Abstract

Machine learning is penetrating most of the classification and recognition tasks performed by a computer. This paper proposes the classification of flower images using a powerful artificial intelligence tool, convolutional neural networks (CNN). A flower image database with 9500 images is considered for the experimentation. The entire database is sub categorized into 4. The CNN training is initiated in five batches and the testing is carried out on all the for datasets. Different CNN architectures were designed and tested with our flower image data to obtain better accuracy in recognition. Various pooling schemes were implemented to improve the classification rates. We achieved 97.78% recognition rate compared to other classifier models reported on the same dataset.

Keywords: *Artificial Neural Networks (ANN); Convolutional Neural Networks (CNN); Deep Learning; Flower Classification; Stochastic Pooling.*

1. Introduction

Image classification is a vibrant area of research in image understanding and computer vision. Abundant classifiers have been proposed in literature for different applications. In this paper, we propose to use convolutional neural networks for classification on flower images. Flowers are a complete natural representation of color, texture and shape features processed under the sun. Flower images are captured with different variations in cameras, angles and lighting conditions.

In our previous works [1] [2] [3], we have performed flower image classification by segmenting the flower images successfully using watershed (WF), marker controlled watershed (MCWF), wavelets (HWF), canny (CF), canny – watershed fusion (CWFF), active contours with shape, color, texture priors (ACSCTF) and fused color – texture featured active contours (ACTFCF). In this work, we propose to eliminate the typical job of segmentation by implementing the classification task directly on raw images using convolutional neural networks.

The flower image dataset from Oxford university of 102 classes is considered for this paper. The sample of dataset is shown in figure 1. In addition, we have created KL University Flower Dataset (KLUFD) of 30 classes. The entire database is sub divided into four datasets based on the complexity in the images. Training is initiated with five different batch sizes. In Batch-I of training dataset 1 is used for training and the validation is carried out on four categories of datasets. Similarly, Batch-II, III, IV of trainings are done using Dataset 2, 3, 4 respectively. Finally, the Batch-V of training is implemented by considering all four datasets for training. Figure 1 and 2 shows the sample data base used for this work. The performance of the CNN algorithms is measured based on their accuracy in recall and recognition rates.

2. Literature review

In [4], a standard visual vocabulary of flower species is created as images in multiple variations under a single flower name as label. The dataset is named as Oxford University flower dataset (OUFD) and on the similar lines we have created a flower dataset named as KL University Flower Dataset (KLUFD). The OUFD consists of 102 species of flower and KLUFD has 32 species of flowers respectively.

The first process in the flower classification problem focuses on image segmentation. In [5] and [6], the authors use a supervised model based flower textures with graph cuts to extract flower content. This process models textures of flower as patch models and the model guides the graph cuts algorithm to initiate the minimum cut at the texture maximum locations in the flower image.

The authors in [7] use color clustering and shape features to perform region of interest (ROI) based flower image retrieval. The color clustering is achieved using color histogram based features. Shape feature set is defined based on centroid contour distance (CCD) and Angle Code Histogram (ACH) characterizing the flower contours. They have tested the algorithm on 885 flower images from 14 species.

A completely unsupervised model is proposed in [8] with simple color based thresholding. RGB color space is converted into Lab color space and OTSU thresholding is performed on all three-color spaces. The best thresholded flower image is selected which is close to ground truth image. The authors claim that their model is faster compared to [5].

In [9], authors use a foreground background model based on Laplacian propagation algorithm computing confidence values of the pixels belonging to foreground or background. The segmentation is tested on 578 flower species with 250000 images and 102 flower species of OUFD with stable segmentation results. The results in [10] are based on the contour matching algorithm of both flower and leaf images. The results are not very encouraging.

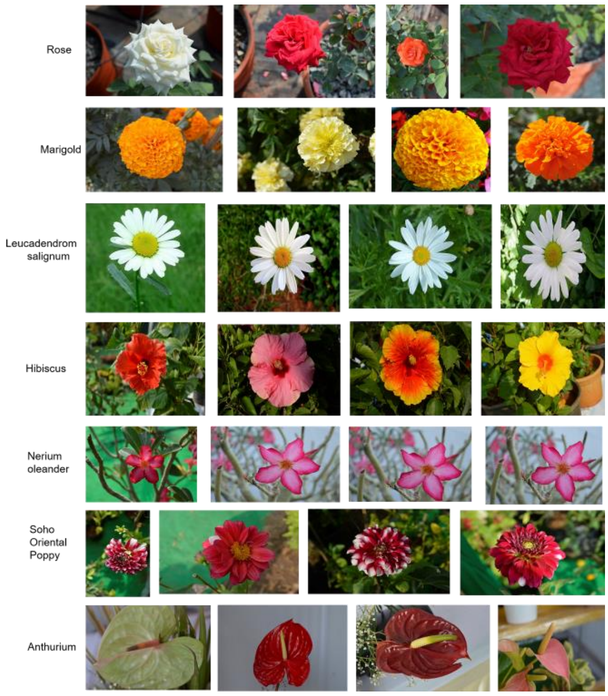


Fig. 1: Sample KL University Flower Dataset (KUFD) With Labels.

The authors in [11] uses an interactive flower segmentation model based on color and shape features. The user must draw a bounding box on the location of flower and the segmentation algorithm uses a flower boundary tracing algorithms extracts the flower regions more accurately. Experiments were conducted on OUFD and results show a near accurate boundary detection on a large set of images.

A large set of features such as color, texture and shapes of flowers is used to classify flower images in [12]. The spatial distribution of features is then classified with support vector machine (SVM) classifier. The testing is initiated on Caltech 101/256 dataset with single feature and multiple feature kernels and found an improvement from 55.1% for single feature to 72.8% for multiple features. Gray level co-occurrence matrix (GLCM) and gabor texture features are combined from a dataset 1250 flower images and classified with KNN classifier in [13]. The flower images are extracted from the internet flower image search. In [14], color and shape features are modelled as a twostep segmentation process. The model is immune to viewpoint changes and petal deformations across different flower classes. The segmentation is produced with Markova random fields (MRF) cost function optimization.

Color image segmentation is used to monitor flower growth in nature with image processing [15]. Flower images used are Lesquerella flowers for oil production. HSI color model with Monte Carlo approach is used for image segmentation. In [16] rose curve based interactive computer visual segmentation model is used effectively for flower classification. Mobile applications are also proposed with android applications for flower classification. Mobile based flower recognition with Difference Image Entropy (DIE) and contour features of the flower from the original flower images. The average recognition is 95% with an average run time 9033ms on mobile platforms [17].

In [18], neural networks are employed on texture features to classify 1800 flower images on 30 varieties. A content based image retrieval system to characterized flower images efficiently. ANN with backpropagation is used as a classifier effectively to recognize flower species. Many hand crafted features were extracted by researchers in classifying the objects using traditional methods.

In recent research, application of deep learning in object recognition is most suitable. CNN is powerful in solving most computer vision based tasks [19-21], [22-23] such as object recognition [24], classification [25]. Classifying at faster rate on a huge dataset is a complicated problem without the knowledge of expert using deep hidden layers CNN extracts image information and avoids the process of complex feature extraction.

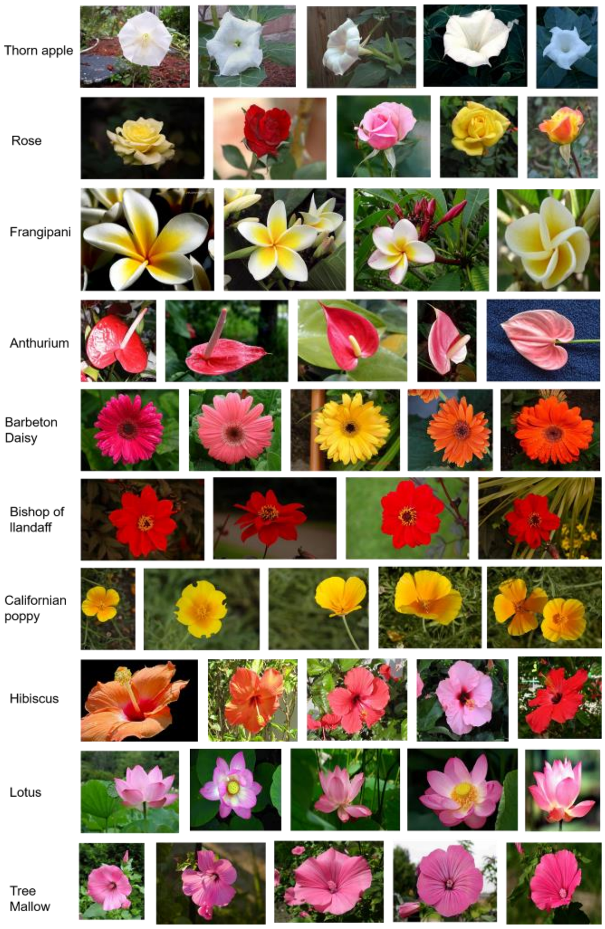


Fig. 2: Sample Oxford University Flower Dataset (OUFD) Flower Images with Class Labels.

Andrew Ng., et al. have performed fundamental research on CNNs to achieve improved performance of CNN algorithms and structural optimization [26-29]. Yann LeCun et al. in [30], highlighted that deep CNN is a breakthrough in image, video, audio and speech processing. So far, no extensive research has done which explores deep CNN for flower classification. The aim of this paper is to bring out the CNN performance in classifying the flower categories effectively.

Deep CNN is suitable for giving solutions to complex problems with huge quantity of data [31]. For example, the classification accuracy is improved in ImageNet dataset [32] which has 1.2 million images almost covering 1000 categories. In such cases we need to consider how to take advantage of CNN. With convolutional neural networks, we need to consider how to design and train a network that adapts to various objects. The major problem to be solved is with the quality and sizes of the images. The unbalanced amounts of low and high quality images in the dataset leads to the unbalanced classification.

The motivation for implementing the deep CNN model for flower classification is that, the feature learning in CNNs is a highly automated from the input images, avoids the complexity in extracting the various features for traditional classifiers. Through the deep architecture, the learned features are deemed as the higher level abstract representation of low level flower images. Hence, we develop the deep CNN model for flower image classification in this paper.

In this paper, a novel CNN based flower classification is proposed to achieve higher recognition rates. Different CNN architectures are implemented, tested on our flower data to bring out the best architecture for classification. Three different pooling techniques namely mean pooling, max pooling and stochastic pooling are implemented and found stochastic pooling is the best for our case. To prove the capability of CNN in recognition, the results are compared with the other traditional state of the art techniques Adaboost, ANN and Deep ANN.

The rest of the paper is as follows: In section 3, the proposed architecture of CNN is described. Section 4 discuss the results obtained in different cases. Finally, section 5 concludes the outcomes of this paper.

3. System architecture

We designed our multi stage CNN model by acquiring knowledge from [33], [34]. The model is constructed with input layer, four convolutional layers, five rectified linear units (ReLu), two stochastic pooling layers, one dense and one SoftMax output layer. Figure 3 shows the proposed system architecture.

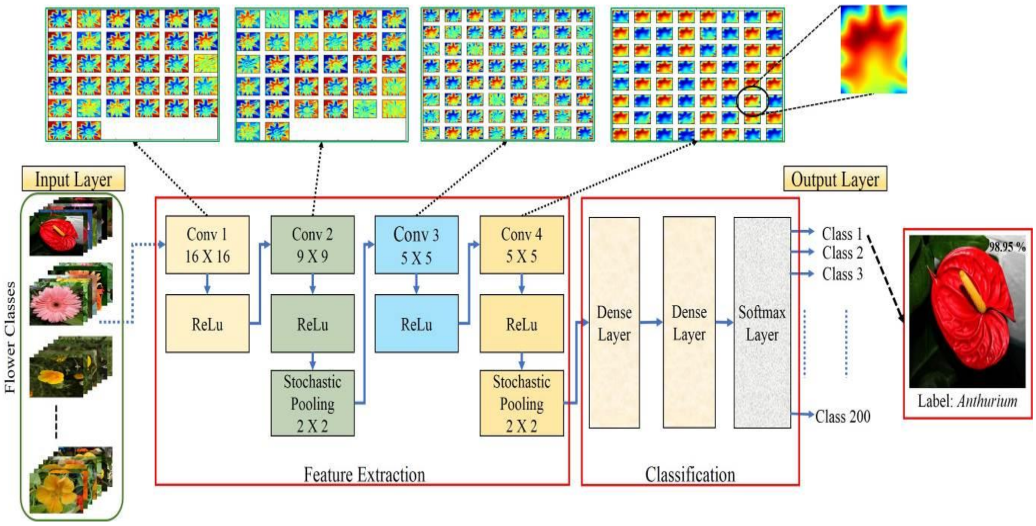


Fig. 3: Proposed Deep CNN Architecture.

The proposed CNN architecture uses four convolutional layers with different window sizes followed by an activation function, and a rectified linear unit for non-linearities. The convolutional windows are of size 16×16, 9×9, 5×5 and 5×5 from layer 1 to 4 respectively. Three kinds of pooling strategies were tested via mean pooling, max pooling, stochastic pooling and found that stochastic pooling is suitable for our application. The feature representation is done by considering two layers of stochastic pooling. Only two layers of pooling is initiated to avoid a substantial information loss in feature representation. Classification stage is implemented with dense/fully connected layers followed by an activation functions. Softmax regression is adopted in classification.

The flower images of size 640×480 are taken as input to the system. As a first step the flower images are pre-processed by resizing them to $128\times128\times3$. Resizing of an input will increase the computational capability of the high-performance computing (HPC) on which the program is being implemented. The HPC used for training the CNN is a 6-node combined CPU-GPU processing machine.

Let us assume an input image of size $I \in R^{w\times h}$. The convolutional kernel with size K is considered for convolution with a stride of and padding for filling the input video frame boundary. The size of the output of convolution layer is given by

$$S_{OUT} = (I - K + 2P)/S + 1 \tag{1}$$

The architecture of our CNN model consists four convolutional layers. While the first two layers extract the low level features (like lines, corners and edges) and the last two layers learn high level features. The detailed layer information and their output sizes with parameters are tabulated in table 1.

The output of a convolutional layer is generally denoted with the following standard equation as:

$$y_j^n = f\left(\sum_{i\in c_j} y_i^{n-1} * k_{ij}^n + \zeta_j^n\right) \tag{2}$$

Where n represents the n^{th} layer, k_{ij} is the convolutional kernel, ζ_j represents bias and the input maps are represented by c_j. The CNN uses a tanh activation function with an additive bias formulated as

$$\hbar_{ni}^{xy} = \tanh\left(\zeta_{ni} + \sum_{w=0}^{w_i-1}\sum_{h=0}^{h_j-1} W_{ij}^{wh}\,\hbar_{i-1}^{(x+w)(y+h)}\right) \tag{3}$$

ζ_{ni} represents feature map bias which are un supervisory trained, w_i, h_j are the kernel width and height respectively. W_{ij}^{wh} is the weight of the kernel at position (w,h). Over a region the max value of a feature is obtained using pooling technique, which reduces the data variance. We implemented our architecture with stochastic pooling technique by calculating the probability values for each region. For every feature map c, the probability is given by

$$\chi_{w,h}^{n,k} = Stochastic_{(w,h,i,j)\in p}\left(\chi_{w,h}^{n-1,k}\ u(i,j)\right) \tag{4}$$

Where $\chi_{w,h}^{n,k}$ is the neuron activation function at a point (w,h) in spatial coordinates, and $u(i,j)$ is the weighing function of window. When compared to other pooling techniques, stochastic pooling makes CNN to converge at faster rate and improves the ability of generalization in processing invariant features.

Table 1: Layer Information and Parameters of CNN

Layer (type)	Function	Output Shape
input		$3\times640\times480$
conv_1	Convolution	$32\times128\times128$
activation_1	Activation	$32\times128\times128$
conv_2	Convolution	$32\times120\times120$
activation_2	Activation	$32\times120\times120$
stoch_pooling_1	Stochastic Pooling	$32\times60\times60$
dropout_1	Dropout	$32\times60\times60$
conv_3	Convolution	$64\times56\times56$
activation_3	Activation	$64\times56\times56$

conv_4	Convolution	64×52×52
activation_4	Activation	64×52×52
stoch_pooling_2	Stochastic Pooling	64×26×26
dropout_2	Dropout	64×26×26
flatten_1	Flatten	43264×1×1
dense_1	Fully connected	64×1×1
activation_5	Activation	64×1×1
dropout_3	Drop out	64×1×1
dense_2	Fully connected	21×1×1
activation_6	Activation	21×1×1
Output	SoftMax Regression	21×1×1

This flower classification task is a multi-class classification problem. Hence, a SoftMax regression layer given by a hypothesis function $h_{\phi}(x)$ is being used as

$$h_{\phi}(x)=\frac{1}{1+e^{\left(-\phi^{T}x\right)}} \tag{5}$$

ϕ must be trained in a way that the cost function $J(\phi)$ is to be minimized.

$$J(\phi)=-\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=0}^{l} l\left\{y^{i}=j\right\}\log_{p}\left(y^{i}=\square\,|\,x^{i};\phi\right)\right] \tag{6}$$

The classification probability in SoftMax regression layer for classifying an input x as a category □ is given as

$$p\left(y^{i}=\square\,|\,x^{i};\phi\right)=\frac{e^{\phi_j^{T}x^{i}}}{\sum_{l=1}^{k}e^{\phi_j^{T}x^{i}}} \tag{7}$$

The network is trained to learn the features of each sign by means of a supervised learning. The internal feature representation reflects the likeness among training samples. We outline 102 flower classes from Oxford University flower dataset (OUFD) and 30 flower classes from our KL University Flower Dataset (KLUFD) in multiple variations. The size of the total dataset is 9500 flower images of 132 classes. All together to know the feature representation learned by the CNN system, the maximized activation neuron is extracted to recognize the sign accurately. Finally, the feature maps were visualized by averaging the image patches with stochastic response in higher layers.

4. Results and discussion

The main goal of this work is to correctly classify the flower image from the OUFD and KLUFD flower dataset. The proposed model of CNN is applied to the flower database for classification. The database consists both OUFD (available online) and KLUFD (created by us) in various orientations. The orientations are due to variations in capture modes. The entire dataset with 132 classes is again divided into 4 sets based on the capturing angle and lighting conditions. Dataset 1 consists single flower in an image with good lighting condition and dataset 2 is having single flower images with poor lighting condition. Flower along with leaves are classified as dataset 3 and images having multiple same flowers are separated as dataset 4. Dataset 3 and 4 contains complex images. Every set from these four datasets contains 132 classes of flower images. Each flower image from each class in the data set is preprocessed by reducing its dimensions to 128×128 which will improve the computational speed of CNN.

4.1. CNN training and testing with different dataset in various batches EXT font of entire document

Training of our proposed CNN model is done in four different batches. In Batch-I of training dataset 1 is used. The images are pre-processed and training is initiated using our proposed CNN architecture. The CNN algorithm is implemented on Python 3.6 platform using a high-performance computing (HPC) machine with 6 CPU-GPU combination.

The CNN is trained using a gradient-descent algorithm at two stages. Stage one handles the multi class classification problem with feedforward pass having training samples from classes. Stage two is the back-propagation pass. The error function is computed as

$$\varepsilon_{e}^{S}=\frac{1}{2}\sum_{m=1}^{S}\sum_{k=1}^{c}\left(l_{k}^{m}-v_{k}^{m}\right)^{2} \tag{8}$$

Where l_{k}^{m} is the label of m^{th} pattern of k^{th} dimension and v_{k}^{m} is the corresponding value of the layer unit. The output of the convolutional layer is the tanh activation function of this value. The back-propagation pass is from higher to lower layers and the error in n^{th} layer is β_{e}^{n} calculated as

$$\beta_{e}^{n}=\left(w^{n+1}\right)^{T}\beta_{e}^{n+1}\,\square\, f^{'}\left(w^{n}y^{n-1}+\zeta^{n}\right) \tag{9}$$

The weight in n^{th} layer is updated according to $\Delta w^{n}=-\lambda\frac{\partial\varepsilon}{\partial w^{n}}$.

During the training different feature maps were observed at different layers. Figure 4 visualizes the feature maps of one flower image obtained in convolutional layer 1 and convolutional layer 2 with 32 filters.

Low level features like lines, edges and corners are learned from Convolutional layer 1 and 2. High level features learned from Convolutional layer 3 and 4 are visualized in figure 5. A stochastic pooling which combines the advantages of both mean and max pooling techniques is implemented. It also overcomes the problem of over fitting. Increasing the number of pooling layers will increase substantial information loss. Hence, the stochastic pooling is implemented in only two layer which is achieved by calculating the probability values of each region.

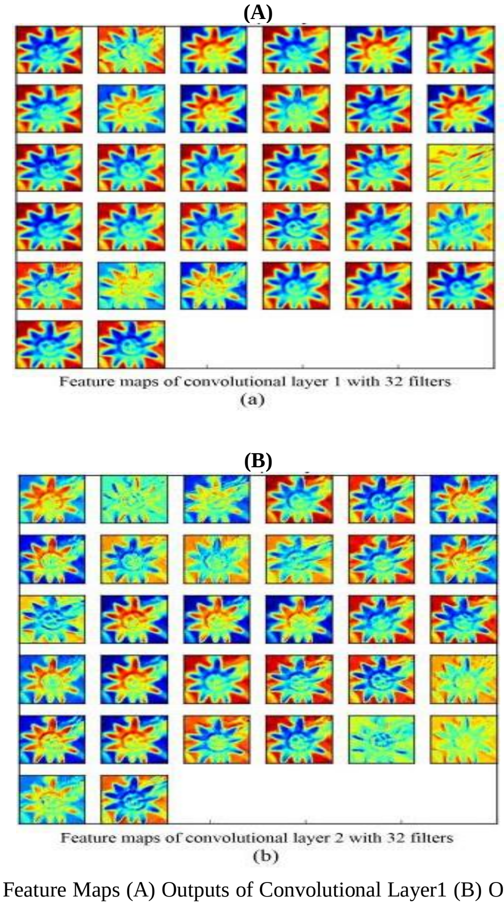

(A)
Feature maps of convolutional layer 1 with 32 filters
(a)
(B)
Feature maps of convolutional layer 2 with 32 filters
(b)

Fig. 4: Feature Maps (A) Outputs of Convolutional Layer1 (B) Outputs of Convolutional Layer2 with 32 Filters Each.

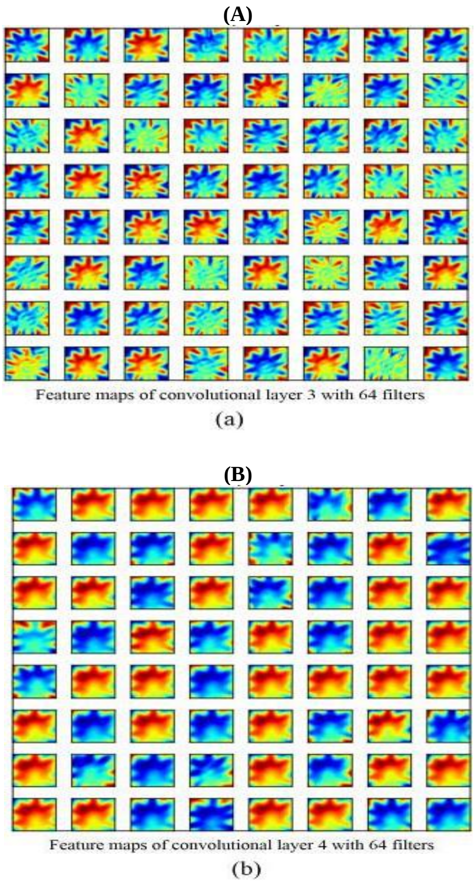


Fig. 5: Feature Maps (A) Outputs of Convolutional Layer3 (B) Outputs of Convolutional Layer4 with 64 Filters Each.

In Batch-I we have used data set 1 for training and testing was carried out on all 4 datasets. In all the cases an acceptable good recognition rates were obtained and are tabulated in table 2.

Table 2: Recognition Rates in Batch-I CNN Training Case.

Training Batch	No. of training data sets	Training datasets	Testing data sets	Recognition rates (%)
Batch-I	1	Dataset-1	Dataset-1	98.34
			Dataset-2	85.06
			Dataset-3	84.12
			Dataset-4	80.82

In Batch-II of training dataset 2 is used. Testing is initiated on all 4 datasets and the classifications rates are tabulated in table 3. As the training data for this case is poor lighting conditioned images, the classification rates are found to be not good.

Table 3: Recognition Rates in Batch-II CNN Training Case.

Training Batch	No. of training data sets	Training datasets	Testing data sets	Recognition rates (%)
Batch-II	1	Dataset-2	Dataset-1	86.19
			Dataset-2	88.12
			Dataset-3	83.96
			Dataset-4	84.65

We have also tested the classification rates in batch-III training of CNN with dataset-3. In this case the flower images are complex due to the leaves. The testing is done and found moderate classification rates as shown in table 4. Further we initiated the batch-IV training with dataset 4 and tested against the all four datasets. Reliable recognition rates were achieved and are tabulated in table 5.

Table 4: Recognition Rates in Batch-III CNN Training Case.

Training Batch	No. of training data sets	Training datasets	Testing data sets	Recognition rates (%)
Batch-III	1	Dataset-3	Dataset-1	91.15
			Dataset-2	89.03
			Dataset-3	94.56
			Dataset-4	92.54

Table 5: Recognition Rates in Batch-IV CNN Training Case.

Training Batch	No. of training data sets	Training datasets	Testing data sets	Recognition rates (%)
Batch-IV	1	Dataset-4	Dataset-1	90.29
			Dataset-2	89.23
			Dataset-3	91.22
			Dataset-4	95.89

Now, all the four datasets are used for training in Batch-V. The classification rates obtained in this case are impressive. Table 6 shows the classification rates on different testing datasets.

Table 6: Recognition Rates in Batch-II CNN Training Case.

Training Batch	No. of training data sets	Training datasets	Testing data sets	Recognition rates (%)
Batch-V	4	Dataset-1 + Dataset-2 + Dataset-3 + Dataset-4	Dataset-1	97.12
			Dataset-2	96.89
			Dataset-3	97.45
			Dataset-4	96.85

Here, by increasing the number of data sets for training it is observed that a good amount of recognition is achieved compared to Batch-I training. It is also observed that the accuracy in recalling the flower is substantially increased as the number of training data sets increased. However, the training time increased by 70% than the Batch-I training process.

Figure 6 shows the training accuracy versus validation accuracy plot for Batch-V training set. It shows that the validation accuracy is good with less amount of over fitting.

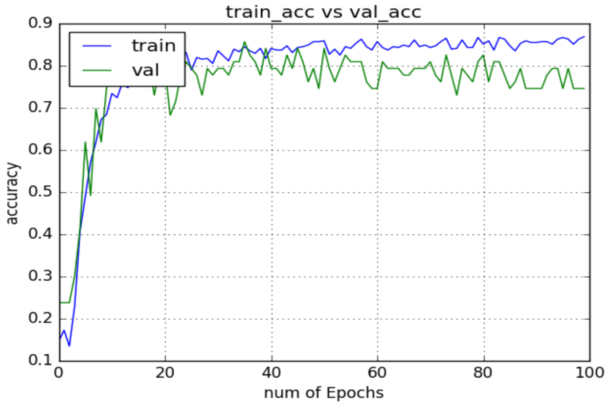


Fig. 6: Training Accuracy and Validation Accuracy.

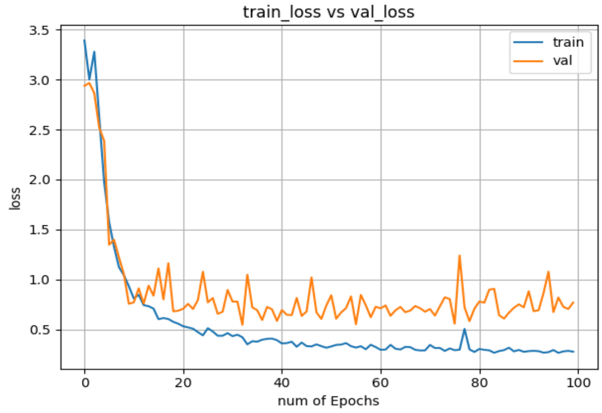
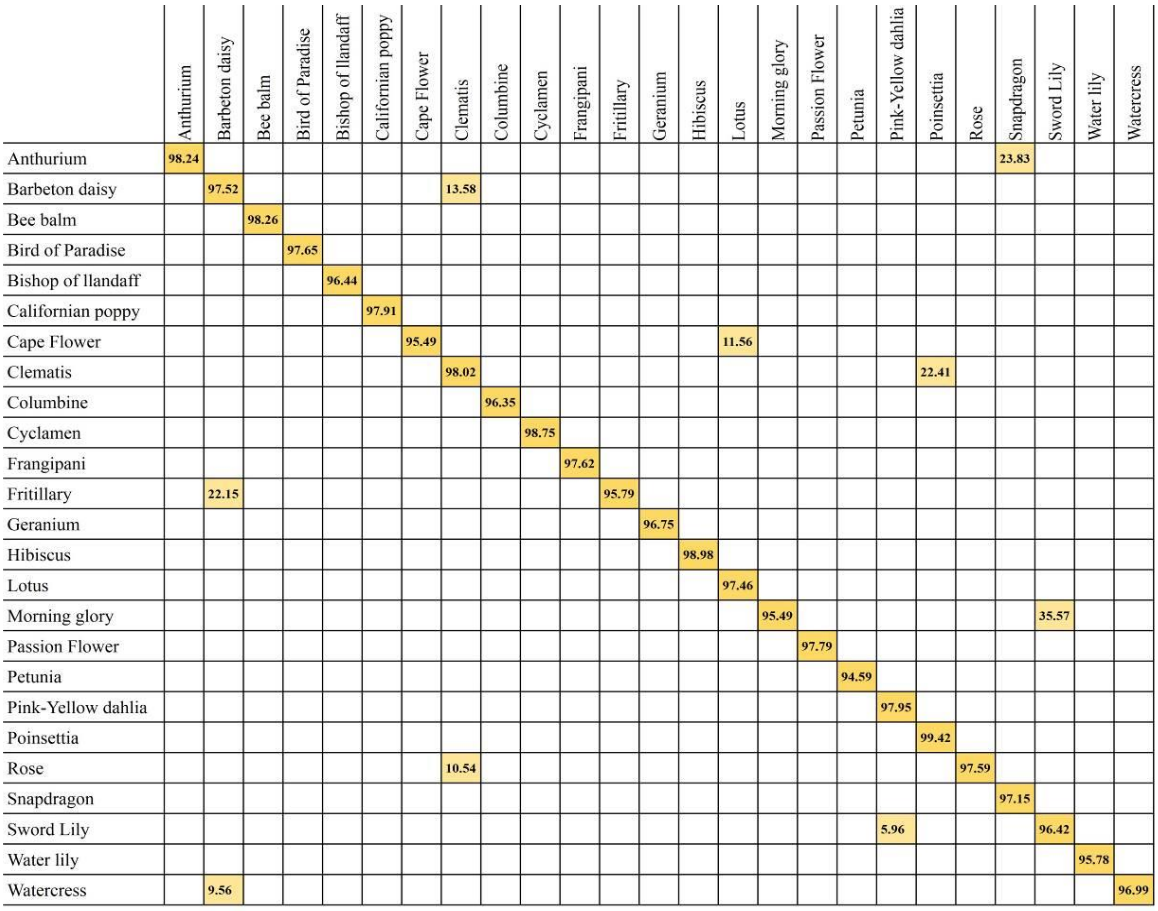


Fig. 7: Training Loss and Validation Loss.

The figure 7, plots losses during training of Batch-II and there is small difference in training and validation losses with an overall less than normal loss coefficient. An average confusion matrix is generated based on the recognition rates and number of matches for training Batch-V is shown in figure 8. For better visualization, it is shown for only 25 flower image classes. However, we sacrifice training computation time for recognition. Time of real time recognition is 0.4 sec per frame and it quite fast compared to algorithms like SVM and Fuzzy classifiers.

All convolutional layers are implemented with different filter windows of sizes 32×32, 16×16, 9×9 and 5×5 . Reducing the filter size improves the recognition rates but increases the computational time due to the increase in number of filters. So, we used convolutional windows of sizes 16×16, 9×9, 5×5 and 5×5 for conv1, conv2, conv3 and conv4 layers respectively. Table 7 compares the performance of choosing different filtering window sizes.



	Anthurium	Barbeton daisy	Bee balm	Bird of Paradise	Bishop of llandaff	Californian poppy	Cape Flower	Clematis	Columbine	Cyclamen	Frangipani	Fritillary	Geranium	Hibiscus	Lotus	Morning glory	Passion Flower	Petunia	Pink-Yellow dahlia	Poinsettia	Rose	Snapdragon	Sword Lily	Water lily	Watercress
Anthurium	98.24																					23.83			
Barbeton daisy		97.52						13.58																	
Bee balm			98.26																						
Bird of Paradise				97.65																					
Bishop of llandaff					96.44																				
Californian poppy						97.91																			
Cape Flower							95.49								11.56										
Clematis								98.02												22.41					
Columbine									96.35																
Cyclamen										98.75															
Frangipani											97.62														
Fritillary		22.15										95.79													
Geranium													96.75												
Hibiscus														98.98											
Lotus															97.46										
Morning glory																95.49							35.57		
Passion Flower																	97.79								
Petunia																		94.59							
Pink-Yellow dahlia																			97.95						
Poinsettia																				99.42					
Rose								10.54													97.59				
Snapdragon																						97.15			
Sword Lily																			5.96				96.42		
Water lily																								95.78	
Watercress		9.56																							96.99

Fig. 8: Confusion Matrices Generated for 25 Flower Classes.

A stochastic pooling adoption attained an average recognition rate of 96.88%. Implementing max pooling and mean pooling produces a recognition rate of 93.33% and 90.84% respectively.

To further know the robustness and efficiency of flower image classification with the proposed CNN, it is compared with other classifiers. For faster recognition, we used Adaboost classifier [40] and ended with a very low classification rates. Further, we replaced Adaboost with a traditional artificial neural network (ANN) [35] [36] [37] [38] for flower image classification and found better recognition rates.

Table 7: Performance Comparison of CNN with Different Convolutional Filter Sizes

	Layers				Recognition Rate (%)	Training Times (Hrs)
	Conv 1	Conv 2	Conv 3	Conv 4		
Convolutional filter window size	16×16	9×9	5×5	5×5	96.88	207
	5×5	5×5	5×5	5×5	97.54	296
	9×9	9×9	9×9	9×9	94.73	205
	16×16	16×16	16×16	16×16	92.15	168
	32×32	32×32	32×32	32×32	90.86	142

Table 8: Recognition Rates with Different Classifiers

Classifier	Recognition Rates (%)					
	Batch-I Training		Batch-IV Training		Batch-V Training	
	Testing with		Testing with		Testing with	
	same dataset	different dataset	same dataset	different dataset	same dataset	different dataset
Adaboost	65.68	60.36	66.47	61.19	67.81	62.91
ANN	79.77	69.68	78.54	71.8	82.45	78.63
DeepANN	87.98	78.89	88.75	81.01	89.74	85.84
Proposed CNN	95.12	89.03	95.89	90.15	97.78	93.98

The recognition accuracy is further improved by replacing ANN with deep ANN [39] and reported an increase in recognition rate by 5%. A much better improvement of 4% in the recognition accuracy and an upward 15% in testing speed were observed in this work with convolutional neural networks. Even though CNN takes more time for training, the testing takes a comparatively far lesser computation times. Classification rates obtained with different classifiers is compared in table 8. Hence, CNN's are a suitable tool for simulating flower classification. Testing is done on a 64 bit CPU with a 4GB ram memory in python 3.6 with OpenCV and Keras Deep learning libraries.

5. Conclusion

CNN is a powerful artificial intelligence tool in pattern classification. In this paper, we proposed a CNN architecture for classifying flower image classes. The CNN architecture is designed with four convolutional layers. Each convolutional layer with different filtering window sizes is considered which improves the speed and accuracy in recognition. A stochastic pooling technique is implemented which combines the advantages of both max and mean pooling techniques. Training is performed in different batches to know the robustness of enormous training modes required for CNN's. In Batch-V of training, the training is performed with four sets of data and maximizing the classification rate. Training accuracy and validation accuracies for this CNN architecture are better than the other models. A less amount of training and validation loss is observed with the proposed CNN architecture. The average recognition rate of proposed CNN model is 97.78 % and is higher compared with the other state of the art classifiers.

References

[1] Inthiyaz, Syed, B. T. P. Madhav, and P. V. V. Kishore. "Flower segmentation with level sets evolution controlled by colour, texture and shape features." Cogent Engineering, vol.4, no. 1, pp.1-15, (2017): 1323572.

[2] Syed Inthiyaz, B. T. P Madhav, P. V. V. Kishore, Vamsi Krishna M., Sri Sai Ram Kumar M., Srikanth K. and Arun Teja B. "Flower Image Segmentation: A Comparison Between Watershed, Marker Controlled Watershed, And Watershed Edge Wavelet Fusion." ARPN Journal of Engineering and Applied Sciences, vol.11, no.15, pp.9382-9387, (2016).

[3] Syed Inthiyaz, B. T. P Madhav, P. V. V. Kishore, "Pre-Informed Level Set for Flower Image Segmentation", International Conference on Smart Computing and Informatics (SCI), March 2017.

[4] Nilsback, M-E., and Andrew Zisserman. "A visual vocabulary for flower classification." In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 2, pp. 1447-1454. IEEE, 2006.

[5] Zhou, Hailing, Jianmin Zheng, and Lei Wei. "Texture aware image segmentation using graph cuts and active contours." Pattern Recognition 46, no. 6 (2013): 1719-1733. https://doi.org/10.1016/j.patcog.2012.12.005.

[6] Nilsback, Maria-Elena, and Andrew Zisserman. "Delving deeper into the whorl of flower segmentation." Image and Vision Computing 28, no. 6 (2010): 1049-1062. https://doi.org/10.1016/j.imavis.2009.10.001.

[7] Hong, An-xiang, Gang Chen, Jun-li Li, Zhe-ru Chi, and Dan Zhang. "A flower image retrieval method based on ROI feature." Journal of Zhejiang University-Science A 5, no. 7 (2004): 764-772. https://doi.org/10.1631/jzus.2004.0764.

[8] Najjar, Asma, and Ezzeddine Zagrouba. "Flower image segmentation based on color analysis and a supervised evaluation." In Communications and Information Technology (ICCIT), 2012 International Conference on, pp. 397-401. IEEE, 2012. https://doi.org/10.1109/ICCITechnol.2012.6285834.

[9] Angelova, Anelia, Shenghuo Zhu, and Yuanqing Lin. "Image segmentation for large-scale subcategory flower recognition." In Applications of Computer Vision (WACV), 2013 IEEE Workshop on, pp. 39-45. IEEE, 2013. https://doi.org/10.1109/WACV.2013.6474997.

[10] Valliammal, N., and S. N. Geethalakshmi. "Automatic recognition system using preferential image segmentation for leaf and flower images." Computer Science & Engineering 1, no. 4 (2011): 13.

[11] Hsu, Tzu-Hsiang, Chang-Hsing Lee, and Ling-Hwei Chen. "An interactive flower image recognition system." Multimedia Tools and Applications 53, no. 1 (2011): 53-73. https://doi.org/10.1007/s11042-010-0490-6.

[12] Nilsback, Maria-Elena, and Andrew Zisserman. "Automated flower classification over a large number of classes." In Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on, pp. 722-729. IEEE, 2008. https://doi.org/10.1109/ICVGIP.2008.47.

[13] Guru, D. S., Y. H. Sharath, and S. Manjunath. "Texture features and KNN in classification of flower images." IJCA, Special Issue on RTIPPR (1) (2010): 21-29.

[14] Nilsback, Maria-Elena, and Andrew Zisserman. "Delving deeper into the whorl of flower segmentation." Image and Vision Computing 28, no. 6 (2010): 1049-1062. https://doi.org/10.1016/j.imavis.2009.10.001.

[15] Thorp, K. R., and D. A. Dierig. "Color image segmentation approach to monitor flowering in lesquerella." Industrial crops and products 34, no. 1 (2011): 1150-1159. https://doi.org/10.1016/j.indcrop.2011.04.002.

[16] Zou, Jie, and George Nagy. "Evaluation of model-based interactive flower recognition." In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 2, pp. 311-314. IEEE, 2004.

[17] Kim, Jung-Hyun, Rong-Guo Huang, Sang-Hyeon Jin, and Kwang-Seok Hong. "Mobile-based flower recognition system." In Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on, vol. 3, pp. 580-583. IEEE, 2009. https://doi.org/10.1109/IITA.2009.407.

[18] F. Siraj, M. A. Salahuddin and S. A. M. Yusof, "Digital Image Classification for Malaysian Blooming Flower," 2010 Second International Conference on Computational Intelligence, Modelling and Simulation, Bali, 2010, pp. 33-38. https://doi.org/10.1109/CIMSiM.2010.92.

[19] Z. Dong, X. Tian, Multi-level photo quality assessment with multi-view features, Neurocomputing. 168 (2015) 308-319. https://doi.org/10.1016/j.neucom.2015.05.095.

[20] Z. Dong, X. Shen, H. Li, X. Tian, Photo quality assessment with DCNN that understands image well, in : Proceedings of the International Conference on MultiMedia Modeling (MMM), 2015, pp.524-535. https://doi.org/10.1007/978-3-319-14442-9_57.

[21] X. Lu, Z. Lin, H. Jin, J. Yang, J. Wang, Rating pictorial aesthetics using deep learning, in: Proceedings of the ACM Conference on Multimedia, 2014, 457-466.

[22] A. Krizhevsky, I.Sutskever, G.E. Hinton, ImageNet classification with deep convolution neural networks, in: Proceedings of the Annual Conference on Neural Information Processing System (NIPS), 2012, pp.1097-1105.

[23] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10,000 classes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1891-1898. https://doi.org/10.1109/CVPR.2014.244.

[24] K. Jarrett, K. Kavukcuoglu, M. Ranzato, Y. LeCun, What is the best multi-stage architecture for object recognition, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2009, pp. 2146-2153. https://doi.org/10.1109/ICCV.2009.5459469.

[25] H. Lee, R. Grosse, R. Ranganath, A.Y.Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, in: Proceedings of the International Conference on Machine Learning (ICML), 2009, pp. 609-616. https://doi.org/10.1145/1553374.1553453.

[26] Y. Bengio, Learning deep architectures for AI, Foundations and trends in Machine Learning, Vol. 2, No. 1, pp. 1-127, 2009. https://doi.org/10.1561/2200000006.

[27] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE , Vol. 86, No. 11, pp. 2278-2324, 1998. https://doi.org/10.1109/5.726791.

[28] H. Lee, A. Battle, R. Raina and A. Y. Ng, Efficient sparse coding algorithms, In Advances in neural information processing systems, pp. 801-808, 2006.

[29] R. Salakhutdinov and G. E. Hinton, Deep Boltzmann Machines, Proceedings of the International Conference on Artificial Intelligence and Statistics, Clearwater Beach, Florida USA, pp. 448-455, 2009.

[30] Y. LeCun, Y. Bengio and G. Hinton, Deep learning, Nature, vol. 521, No. 7553, pp. 436-444, 2015. https://doi.org/10.1038/nature14539.

[31] H. Lee, R. Grosse, R. Ranganath, A.Y.Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, in: Proceedings of the International Conference on Machine Learning (ICML), 2009, pp. 609-616. https://doi.org/10.1145/1553374.1553453.

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: a large-scale hierarchical image dataset, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248-255.

[33] A. Krizhevsky, I. Sutskever and G. E. Hinton, Imagenet classification with deep convolutional neural networks, in Advances in Neural Information Processing Systems(NIPS), Lake Tahoe, Nevada, USA pp. 1097-1105, 2012.

[34] Y. LeCun, Y. Bengio and G. Hinton, Deep learning, Nature, vol. 521, No. 7553, pp. 436-444, 2015. https://doi.org/10.1038/nature14539.

[35] Rao, G. Anantha, P. V. V. Kishore, D. Anil Kumar, and A. S. C. S. Sastry. "Neural network classifier for continuous sign language recognition with selfie video." Far East Journal of Electronics and Communications 17.1 (2017): 49.

[36] Kishore, P. V. V., M. V. D. Prasad, Ch Raghava Prasad, and R. Rahul. "4-Camera model for sign language recognition using elliptical fourier descriptors and ANN." In Signal Processing and Communication Engineering Systems (SPACES), 2015 International Conference on, pp. 34-38. IEEE, 2015.

[37] Kishore, P. V. V., A. S. C. S. Sastry, and A. Kartheek. "Visual-verbal machine interpreter for sign language recognition under versatile video backgrounds." In Networks & Soft Computing (ICNSC), 2014 First International Conference on, pp. 135-140. IEEE, 2014. https://doi.org/10.1109/CNSC.2014.6906696.

[38] Kishore, P. V. V., S. R. C. Kishore, and M. V. D. Prasad. "Conglomeration of hand shapes and texture information for recognizing gestures of Indian sign language using feed forward neural networks." International Journal of engineering and Technology (IJET) 5, no. 5 (2013): 3742-3756.

[39] Rao, G. Ananth, and P. V. V. Kishore. "Selfie video based continuous Indian sign language recognition system." Ain Shams Engineering Journal (2017).

[40] K. V. V. Kumar, P. V. V. Kishore, and D. Anil Kumar, “Indian Classical Dance Classification with Adaboost Multiclass Classifier on Multifeature Fusion,” Mathematical Problems in Engineering, vol. 2017, Article ID 6204742, 18 pages, 2017. https://doi.org/10.1155/2017/6204742.