A Hybrid Methodology for Flower Images Segmentation & Recognition with extended Deep-Convolution Neural Network (CNN)

Neeraj Kumar Rathore (✉ neerajrathore37@gmail.com )
IGNTU: Indira Gandhi National Tribal University  https://orcid.org/0000-0002-9349-3419

Varshali Jaiswal
Rajiv Gandhi Technical University: Rajiv Gandhi Proudyogiki Vishwavidyalaya

Varsha Sharma
Rajiv Gandhi Technical University: Rajiv Gandhi Proudyogiki Vishwavidyalaya

Sunita Varma
Shri Govindram Seksaria Institute of Technology and Science

Research Article

Keywords: Image segmentation, Deep convolution neural network, Classification, Edge detection, Computer Vision, L*a*b color space

DOI: https://doi.org/10.21203/rs.3.rs-621258/v1

License: © This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License
A Hybrid Methodology for Flower Images Segmentation & Recognition with extended Deep-Convolution Neural Network (CNN)

Neeraj Kumar Rathore¹, Varshali Jaiswal², Varsha Sharma³, Sunita Varma⁴
¹Associate Professor, Indira Gandhi National Tribal University (IGNTU)-A Central University, Amarkantak, M.P.
²Research Scholar, School of Information Technology, RGPV, Bhopal, India
³Assistant Professor, School of Information Technology, RGPV, Bhopal, India
⁴Professor, Information Technology, Shri. GSITS, Indore, India

Abstract— Deep-Convolution Neural Network (CNN) is the branch of computer science. Deep Learning CNN is a methodology that teaches computer systems to do what comes naturally to humans. It is a method that learns by example and experience. This is a heuristic-based method to solve computationally exhaustive problems that are not resolved in a polynomial computation time like NP-Hard problems. The purpose of this research is to develop a hybrid methodology for the detection and segmentation of flower images that utilize the extension of the deep CNN. The plant, leaf, and flower image detection are the most challenging issues due to a wide variety of classes, based on their amount of texture, color distinctiveness, shape distinctiveness, and different size. The proposed methodology is implemented in Matlab with deep learning Tool Box and the dataset of flower image is taken from Kaggle with five different classes like daisy, dandelion, rose, tulip, and sunflower. This methodology takes an input of different flower images from data sets, then converts it from RGB (Red, Green, Blue) color model to the L*a*b color model. L*a*b has reduced the effort of image segmentation. The flower image segmentation has been performed by the canny edge detection algorithm which provided better results. The implemented extended deep learning convolution neural network can accurately recognize varieties of flower images. The learning accuracy of the proposed hybrid method is up to 98% that is maximizing up to +1.89% from state of the art.

Keywords — Image segmentation, Deep convolution neural network, Classification, Edge detection, Computer Vision, L*a*b color space.

I. INTRODUCTION

Ayurveda prompts plant, leaf, and flower image segmentation and recognition. Image segmentation refers to separating an image into multiple areas [1]. The objective of segmentation is to make things easy and refine the representation of an image into something that is more expressive and easier to analyze. Image segmentation is normally used to trace the boundaries (lines, curves, etc.) and objects in the image. Edge detection is the segmentation process which identifies the edges of an image. There are various algorithms for edge detection like canny edge detection, watersheds, etc.

The flower image segmentation and recognition are difficult tasks caused by the broad assortment of flower image types, based on their amount of texture, color distinctiveness, shape distinctiveness, and different size. Various categories of the flower image are similar, as shown in figure 1, both in shape or color because of inter-class similarity.
The deep convolution neural network is used as a tool for image processing to solve various problems [2] such as object classification, object detection, face recognition, and semantic sub-splitting of the image, etc. In semantic image segmentation, the convolution neural network takes an image as input. After processing the input image it provides output as sub-splitting of the original image where each class is identified with mathematical value and color. The classes are demarcated by the user and speak of the diverse objects in the image like floors, dogs, and chairs. The intermediate layers are associated with learnable weights. The neural network has some significant parameters denoted as hyperparameters. The learning rates, batch size, number of the intermediate layer are some examples of hyperparameters.

**Problem Definition** — In Ayurveda various types of flowers like rose, bhringraj, hibiscus, etc are used for making medicines. Ayurveda prompts to identify and recognize the flower images. But identifying and recognizing an image is a very difficult task because different species of flowers may look identical in shape as well as in color. It is a very challenging task to maximize accuracy and minimize the error rate of image segmentation and detection of RGB flower images. Similarly, RGB images are considered five-dimensional problems. Since three dimensions are used for red, green, blue. The two dimensions are used for the luminosity layer and chromaticity layer which as solved by L*a*b color space. It is also considered as a complex computational time-consuming problem.

To solve this problem proposed system utilize an optimization method of deep convolution neural network for identifying and segmenting flower image and L*a*b conversion. It is a fast and accurate image segmentation method. In summary, contributions of the proposed research methodology are as follows.

1) Developed and implemented a hybrid method that reads the number of N different colored flower image from data sets.

2) Every color image has five attributes, first three for colors Red, Green, Blue, and two attributes for the geometry luminosity layer and chromaticity layer. To provide better results and minimize the effort of image segmentation every image is transferred into L*a*b color [2] [3].

3) Then resultant uniform size flower images are segmented using a canny edge detection algorithm which becomes an essential field of research work to improve the experimental results and identification of various objects in image [4].

4) The segmented flower images are grouped into different cluster. In a cluster, each pixel is similar according to some properties such as color, texture, and intensity.

5) To design an extended deep convolution neural network from resultant image datasets with one hidden layer for prediction of flower images. The deep convolution neural network is structured in computational layers interchanging amongst the convolution layer and max-pooling layer.

6) In the last predicted various flower images with proposed hybrid methodology on flower image datasets.

7) The proposed methodology would be compared with other pre-trained convolution neural networks such as VggNet, GoogLeNet, Alexnet, etc [5].

In this paper a novel extended hybrid deep learning convolution neural network is developed and implemented for flower image detection and segmentation. In the proposed algorithm an RGB image is first converted into L*a*b image to reduce the efforts in image segmentation and provide effective results. Then an extended deep CNN is trained for detection of flower images with one hidden layer. In the end, the various flower images are recognized by extended deep CNN and the results are calculated. New better flower image segmentation schemes are thus derived, which overcome the problems of conventional systems, by maximizing learning accuracy up to 98% i.e. increasing +1.89% from state of the art and should substantiate augmented flower image segmentation problems.

The remaining sections of the paper are structured as follows: Literature review of research work is highlighted in Section 2. The proposed methodology is illustrated in Section 3. Section 4 shows the experimental setup. Section 5 gives results analysis. In the end, the proposed work is concluded in Section 6 based on the experimental results.

### II. LITERATURE REVIEW

Flower image detection, segmentation are essential and challenging task in Ayurveda. Ayurveda uses various types of flower like Marigold, Hibiscus, Sunflower, Lavender, Echinacea, Chamomile, Yarrow, Cornflower, Rose, Bhringraj, etc in medicines. But it is a difficult task as the different species of flowers may look very identical, both in shape or color inter-class.

Many researchers are working on flower image detection and segmentation. Table 1. gives a comparison study about related works with performance criteria.
### Table I
Comparison of Related Works About Image Segmentation, Reorganization and Performance Criteria

| Row | Authors | Location of study | Type of data sets | Date of research | Method |
|-----|---------|-------------------|-------------------|------------------|--------|
| 1   | Kristen et al. [11] | USA | Image Datasets/Caltech101 | 2005 | * |
| 2   | Jonathan Long [17] | UC Berkeley | PASCAL VOC 2011 and 2012, ImageNet Large-Scale Visual Challenge (ILSVRC) | 2011 | * |
| 3   | Krizhevsky [15] | Canada | Recognition Challenge (ILSVRC) | 2012 | * |
| 4   | Jeff Donahue [29] | USA | SUN-397 database | 2014 | * |
| 5   | Karen Simonya et al. [23] | UK | ILSVRC-2012 dataset | 2014 | * |
| 6   | Ross Girshick [13] | UC Berkeley | VOC 2007 dataset | 2014 | * |
| 7   | Qing Li et al. [18] | United States | ILD database | 2014 | * |
| 8   | Sergey Ioffe [19] | USA | GoogleLeNet ensemble ILSVRC-2012 dataset | 2015 | * |
| 9   | Karen Simonya et al. | UK | ILSVRC-2012 dataset | 2015 | * |
| 10  | Xiaofeng Zhu et al. [25] | USA and Korea | MCI-NC | 2015 | |
| 11  | Kaiming He et al. [22] | USA | ILSVRC & COCO | 2015 | |
| 12  | Zhao & Li [6] | Hayit Greenspan et al. [14] | - | 2016 | |
| 13  | Qi Dou et al. [16] | Hong Kong | SWI-CMB | 2016 | |
| 14  | Ling Zhang et al. [21] | China | Herlev Dataset | 2017 | |
| 15  | Nitin R. et al. [7] | India | Oxford-102 flower dataset | 2017 | |
| 16  | Xiaoling Xia et al. [8] | China | Oxford-17 flower dataset | 2017 | |
| 17  | I.Gogul et al. [9] | India | Oxford 17 | 2018 | |
| 18  | Hazem et al. [6] | UK | PSREU, USA | 2019 | |
| 19  | Yang Chen et al. [5] | USA | Kaggle datasets | 2020 | |
| 20  | Current work | India | Kaggle datasets | 2020 | |

(Continued)
This section reviews some state arts in the area of flower image segmentation. Hazem et al. proposed a flower image segmentation, classification, and detection method using the deep learning method [6]. This approach tested the three well-known datasets and outcomes were equated with supplementary methods. The accuracy of the proposed methods is up to 97%. The proposed approach of flower images segmentation used localization of the minimum bounding box around images and then utilized binary classification in a convolution neural network for detection of flower images.

Rashmi et al. proposed a new architecture of the convolution neural network to implement mobile Net applications for the classification of flower images by reducing the size and latency of original images. The application was small, efficient, and optimized the results [7]. The experiment was examined on the TensorFlow platform and results shows system required minimize time and space of flower images classification.

A flower classification approach proposed by Xiaoling et al. used the Inception-v3 model of the TensorFlow platform [8]. This flower image classification technique utilized the transfer learning method. The accuracy of flower image classification was up to 95% on Oxford 17datasets and 94% on 120’ flower datasets.

A new Convolution Neural Network approach proposed by I.Gogul et al. trained the small data sets and utilized general purpose computation resources [9] the approach classified the flower image into three different parts and then extracted the features of flower images by using OverFeat CNN in training data sets. The proposed system utilized machine learning techniques for training a neural network.

This proposed system had accuracy up to Rank-1, 82.32% and Rank-5 accuracy of 97.5% using machine learning classification FLOWERS 28 dataset [10]. The researchers proposed a flower recognition approach with Support Vector Machine classification [11]. This approach used natural flower images for image segmentation purpose and learning accuracy of the system was up to 85.93%.

1.1 Image Segmentation

The image segmentation method improves the performance of object recognition and identification. Image segmentation is a technique of partitioning an image into a segment that is similar according to sets of predefined criteria. Semantic
segmentation faces an inherent tension between semantics and location [12]. Region-based classification is a standard technique for semantic segmentation that can be easily applied on flower image segmentation [13]. Flower image segmentation is the technique of detecting edges of flower image that make into a different group. Pixels or spots in a group are similar according to some properties of a group such as color, texture, and intensity.

1.2 L*a*b color Space
The color model based on L*a*b space is q layered design method in which the red-green axis denoted by chromaticity layer 'a*'. Similarly, blue-yellow axis is denoted by chromaticity layer 'b*' and luminosity is denoted by 'L*' layer. The color image has five attributes, first three attributes for the color red, green, blue, and two attributes for the geometry luminosity layer and chromaticity layer. To provide a better result and minimize the effort of image segmentation, every image is transferred into L*a*b color model. The conversion between the RGB color model and the L*a*b* color model is not a straight forward process.

1.3 Artificial Deep learning Convolution Neural Network
The biological neural network is the inspiration of convolution neural network [14]. Artificial deep learning-based convolution neural network is a mathematical model. Mathematical models are weighted function, input function and transform function [15]. Figure 2 represents the architecture of artificial deep convolution neural network, where R is size of input and corresponding output.

\[ n = W_{1,1}P_1 + W_{1,2}P_2 + \ldots + W_{1,R}P_{R+b} \]  

Where f represents mathematical model, a represented activation function, w represents a weight function, b represents a biased value, and p represents input value.

Researchers researched in the area of deep learning for recognizing and segmentation of images. Pheng et al. presented an improved technique to automatically identify cerebral microbleeds from magnetic resonance volumes, leveraging 3D convolution neural network [16]. The proposed system utilized a 3D convolution neural network for recognition of cerebral microbleeds for magnetic resonance images. The authors compared experiment results with previous method that was based on the 2D convolution neural network. The proposed method had enhanced recognition accuracy compared to other methods. This method used for image recognition and sub-splitting tasks with an advanced application of a 3D convolution neural network for volumetric medical datasets.

The authors Jonathan et al. implemented fully convolution neural networks for semantic segmentation problem and improved the architecture with a multi-resolution layer [17]. Fully convolution neural networks take input of arbitrary size and produce output with the corresponding size. The authors utilized Alex Net, VGG Net, and GoogleLeNet with fully convolution neural networks to generate efficient and meticulous semantic segmentation of images. This combination dramatically improved the state-of-the-art, while simultaneously simplified and speeded the learning rate and inference.

Qing et al. proposed a customized convolution neural network architecture to categorize HRCT lung picture patch of ILD pattern [18]. The system designed a convolution neural networks with a single layer that learns the DCT pattern from training samples capably and produced efficient classification results. The experiment results show that the proposed approach is capable of extracting discriminative features automatically and achieve efficient target performance. In the experiment observed that the limited size of training data and blurry structure were complicated in adjusting the convolution neural network to ILD image classification.

Deep convolution neural networks were produces efficient results in the field of image recognition, categorization with a large size of data sets [19] [20]. Several researchers are working to achieve efficient classification results or
recognition of objects accurately without considering the computation complexity [21] [22] [23]. The additional hardware requirement of the convolution neural network is graphics processing units or titanium processor to process large datasets.

1.4 Classification

Classification of flower image is a challenging task due to the intra-class variability with different lighting conditions. The proposed flower image segmentation method used a binary classification technique to detect flower images. The use of extended deep convolution neural networks [5] is for classification problems where the output to an image is an alone class label [24]. However, in many computer vision problems, particularly in biomedical image processing the desired output should include localization [25] [26].

It is similar to flower image classification because of varieties of flower images, having similarities in various groups. Satoshi et al. implemented a color classification of an object that utilized three heterogeneous co-occurrence colors such as CoHD, CoHOG, CoHED [27]. The results produced by that method efficiently achieve a high classification rate due to high dimensional and highly discriminating co-occurrence features.

III. PROPOSED METHODOLOGY

This section proposed, a hybrid methodology for identifying and splitting up of flower image using the L*a*b color space with an extended deep convolution neural network. The improvement of artificial neural networks is done by a deep learning approach. Deep learning having supplementary layers that authority higher levels of abstraction and increasing objects detection rate. Figure 3 describes the proposed hybrid methodology for flower image segmentation and recognition with L*a*b color space by an extended deep convolution neural network. The proposed algorithm work as follows:

Step 1: N number of different flower images are read from the data sets in which each image has five attributes, three attributes of color (Red, Green and Blue) and two attributes for geometry (luminosity layer and chromaticity layer). The conversion between RGB to L*a*b color space [3] improved the performance of image segmentation and produced excellent results.

Step 2: Then resultant uniform size flower images are segmented using a canny edge detection approach which becomes an essential field of research work to improve the experimental results and identify various objects in images. Flower image segmentation is the technique of detecting edges of flower images that make into different groups. Pixels or spots in a group are similar according to some properties of a group such as color, texture, and intensity. Figure 4 shows below is results of conversion between RGB to L*a*b color space with canny edge detection flower image segmentation downloaded from the experiment.
Step 3: An extended deep convolution neural network is then trained from flower image data sets. There are various ways to modify the deep convolution neural network like altering the input image type, sliding stride, filter size, and filter types. In the proposed methodology modified input layer. The input layer reads L*a*b image instead of the RGB image for learning.

1) The first image input layer, grips the processed pixel value of the image in which image is represented by height and width with dimensions of the color for normalization of an input image.

2) The second convolution layer calculates the results of neurons that are associated with restricted areas in the participation. Each node calculated scalar multiplication between their weights and input value which are associated with the input region.

These may consequence in size and figure 5 represents the convolution layer. This figure is taken an input array of a pixel with different kernels size and generates an output pixel in the form of an array.

Layer of a fully connected convolution neural network described by equation 2:

$$z = \sigma(Wx + b)$$  

Here, $x$ represents input vector, $z$ represents output vector, $W$ represents weight matrix, $b$ represents bias vector and $\sigma$ represents activation function. The figure 6 described working on convolution layer which is the output of dot product of two matrixes: one is image matrix and the second is filtered (kernel) matrix with different load values [28].

For example, a convolved feature value equal to 4 is calculated by the dot product of the filter matrix and image matrix, $1*1+0*0+1*0+0*1+1*1+0*0+1*1+0*1+1*1 = 4$ and the same calculation is repeated for all regions. Convolution layer is related to convolution of two signals represented by the equation 3:

$$f_1[x, y] * f_2[x, y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f_1[n_1, n_2] * f_2[x-n_1, y-n_2]$$

(3)

3) The threshold operation is performed by third layer called ReLU layer. In ReLU layer, any value of weight less than zero are assigned to zero.

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

$$f(\text{net}) = \max(0, \text{net})$$  

(4)

The ReLU layer not changed the size of volume that is and produces an activation function that is max (0, x) with a threshold value at zero. Equation 3 show activation function with input value x and corresponding output function f. ReLU considered as the nonlinearity in the convolution neural network.

Dimensions of output volume $= \frac{[W-F+2P]}{S} + 1$  

(5)

The dimensions of output volume described by equation 5, in which W represented input volume, F represented size of the filter, where P represented the amount of padding, and S represented the stride. Figure 7 show ReLu layer used by the proposed method.

4) Fourth Max-pooling layer accomplishes a behind sample process alongside the spatial dimension height and width consequential in a dimension similar to $[16x16x12]$. Likewise, if pool-Size is $[2, 3]$, then the maxpool layer calculates average value of region of height 2 and width 3.
Figure 8 shows a matrix representation of maxpooling layer in the convolution neural network.

\[
\text{Size of output dimension} = \frac{\lfloor W - F \rfloor}{S} + 1
\]  \hspace{1cm} (6)

5) In the proposed method used one intermediate layer (hidden layer) which is again a combination of the entire above layer.

6) The fifth layer is a fully-connected layer in which those neurons are considered that are fully connected in the Max-pooling layer. It is a fully connected layer for convolution neural network with activation function like Soft-Max for various classification approaches. It produces resultant output with dimension, volume \([1*1*N]\) where \(N\) represents the number of classes for classification.

7) Multi-class classification problems solved by calculating cross-entropy loss function on the classification layer by mutually exclusive classes.

Step 5: Different varieties of flower images are classified from datasets.

Step 6: A graph is plotted between the number of iterations and accuracy, and similarly between the number of iterations and error rate.

The proposed method is simulated with the experiments. The objective of testing is towards improved training accuracy and minimizing the error rate for different flower images using extended deep learning convolution neural network-based algorithm [29].

IV. EXPERIMENT SETUP

The proposed method is simulated and results are analyzed for the publicly available five flower families taken from Kaggle. The five flower families are Daisy, Dandelion, Rose, Sunflower, and Tulip. The proposed extended hybrid methodology is experimented with coloured flower images from Kaggle’s flower image data sets with size of 512*512 pixels [30]. In this experiment system uses 80% images for training and 20% images for testing. The Figure 9 shows some flower images from different families of flowers from Kaggle’s flower image data sets.

1.5 Evaluation Metrics

There are a number of parameters used to calculate the accuracy of flower image classification, segmentation, and detection in the suggested method.

\[
\text{Accuracy} = \frac{\text{Sum}(Y_{\text{Test}} = T_{\text{Test}})}{\text{Numel}(T_{\text{Test}})}
\]  \hspace{1cm} (7)

\[
\text{Error rate} = 1 - \text{Accuracy} = \frac{\text{Sum}(Y_{\text{Test}} \neq T_{\text{Test}})}{\text{Numel}(T_{\text{Test}})}
\]  \hspace{1cm} (8)

Accuracy is measured as the ratio of the count of accurate labels inside the test flower image datasets corresponding to
the classification from classify, to the count of images in the test flower images.

1.6 Pseudo code

Algorithm: DCNNFCD: Deep learning convolution neural network for flower image classification and detection

Input: NOI: Number of flower images
im: input the RGB color flower images

Output: Trained neural network for flower image classification and reorganization.

1: for block:= 1 to NOI
2: Read flower image
3: Resize flower image
4: Convert the RGB images in l*a*b space.
5: Apply canny edge detection algorithm to image for texture feature
6: Combined the a*b images with edge image
7: Repeat step 2 to 5, for each image and collect image into one array.
8: Apply the CNN on the image array with defined classes.
10: Train the CNN with number of iterations = 20 and initial learning rate 0= 0.0001
11: Similarity test image data sets are also prepared. This data is proposed and tested using trained CNN.
12: The Accuracy, Computation time, Error rate has been calculated and displayed.

1.7 Implementation Details

The proposed system experimented in two different sections, Experiment_1 and Experiment_2 with different hardware configurations and the different number of processed images. In Experiment_1, minimum hardware configuration and only 100+ images were used whereas in Experiment_2 extra hardware GPU with 32GB RAM and 1000+ images were used.

**Experiment_1**

In this experiment, 5 to 20 different images, each of five flower families (Daisy, Dandelion, Rose, Sunflower, and Tulip) were used. During the experiment_1, 100 images were used by the proposed method from the database.

**Implementation Details for Experiment 1**

| S.NO | Layer                  | Parameter                  |
|------|------------------------|----------------------------|
| 1    | Convolution Layer 1-2  | 5, 20, stride=1            |
| 2    | Max pool               | 3, stride=2                |
| 3    | Convolution Layer 3-4  | 3*3, 64, stride=1          |
| 4    | Max pool               | 2*2, stride=2              |
| 5    | Fully Connected Layer   | 5, // output size          |

In experiment 2, 165 different images of each type of flowers used, so that total of 825 images was processed in the database.

**Implementation Details for Experiment 2**

The implementation for experiment_2 was done in Matlab 2019a using deep learning Tool Box and results were calculated on an Intel(R) Xeon(R) W-2133 cpu@3.60GHz.
32.0 GB RAM running window 10 Pro for Workstations and 64-bit Operating System, x64-based processor. Simulation parameter used by the experiment-2 was same as experiment_1 which is shown in Table 2. The flower image segmentation and classification utilizes extended convolution neural network in which first all images are resized to $64 \times 64 \times 3$ to provide a uniform and normalized combination of images that pass through the convolution neural networks.

NOI (Number of images of each class) = 165 and Training on single GPU.

Initializing input data normalization.

2. RESULTS ANALYSIS

Results Analysis of Experiment 1

Experiments were conducted over downloaded flower image datasets from the website Kaggle. In Table 3 the first column shows the NOI (number of images of each type of flower and there are five types of flower used by the proposed algorithm for simulation), the second column represents epoch, and the following columns represent the time elapsed, loss, accuracy, and learning rate respectively, used by the proposed algorithm. The training was done on a single CPU by initializing flower image normalization by $256 \times 256 \times 3$.

| NOI | Epoch | Time Elapsed (Seconds) | Mini-batch Loss | Mini-batch Accuracy | Base-Learning Rate |
|-----|-------|-------------------------|-----------------|---------------------|--------------------|
| 4   | 1     | 8.00                    | 6.2823          | 15.00%              | 1.00e-04           |
|     | 20    | 137.57                  | 0.0002          | 100.00%             | 1.00e-04           |
| 8   | 1     | 13.61                   | 7.8754          | 20.00%              | 1.00e-04           |
|     | 20    | 266.59                  | 0.0353          | 99.00%              | 1.00e-04           |
| 12  | 1     | 20.45                   | 6.0824          | 21.67%              | 1.00e-04           |
|     | 20    | 404.36                  | 0.0383          | 98.73%              | 1.00e-04           |
| 16  | 1     | 28.53                   | 6.9970          | 23.75%              | 1.00e-04           |
|     | 20    | 533.66                  | 0.0409          | 98.35%              | 1.00e-04           |
| 20  | 1     | 62.21                   | 7.7679          | 18.00%              | 1.00e-04           |
|     | 20    | 711.51                  | 0.7370          | 97.75%              | 1.00e-04           |
| 24  | 1     | 126.00                  | 6.4503          | 22.50%              | 1.00e-04           |
|     | 20    | 6249.48                 | 0.9507          | 95.00%              | 1.00e-04           |

The proposed algorithm was experimented with different number of flower images and maximum epoch was 20. From the results, proposed method shows that when NOI = 4 and epoch is equal to 20, then the time required for training is 137.57s and accuracy is 100% with minimum batch loss equal to 0.0002 and learning rate is 1.00e-04. Figure 10 gives an average of the computation time in seconds taken by over the proposed algorithm for training flower image with NOI (number of images). But when NOI=20, total images=40, then at epoch =20, the time required for learning rate is 266.59s and accuracy decreased only 95.00% with loss 0.1353.
Figure 10: The Average of the computation time in seconds taken by the proposed algorithm for training flower image with NOI (number of images).

Again, when NOI = 12, 16, 20, 24 and total flower images are equal to 60, 80, 100, 120 respectively the accuracy is 98.33%, 98.75%, 88.00%, 95.00% and times is 404.36s, 533.66s, 711.51s, 6249.48s with same learning rate.

Figure 11 shows the graph between NOI (number of images) with the average accuracy over the proposed algorithm.

The proposed algorithm achieved an average accuracy of 95.84% and an average loss rate of 0.1837. Figure 12 shows graphically, the relation between NOI (number of images) and the average minimum–batch loss over the proposed algorithm.
The percentage of wrong detection of flower is still high and needs further improvement. The proposed deep learning-based convolution neural network [32-48] provides best flower classification results on all datasets.

### TABLE IV

| Types of Flower | Time required for Flower classifier |
|-----------------|-------------------------------------|
| Tulip           | 16.183s                             |
| Rose            | 13.989s                             |
| Dandelion       | 10.101s                             |
| Daisy           | 10.101s                             |

Fully convolution neural networks is a rich class of models in which modern classification convolution neural network is a special case. Figure 13 (a, b, c, d) shows detected flower image which was downloaded from mathwork group during the experiment. Figure 13(a), 13(b), 13(c), 13(d) represented detection of Rose, Tulip, Dandelion, and Daisy flower image simultaneously.
Results Analysis for experiment_2

The training results are shown in Table 5. The columns represent the number of Epoch, iteration, time elapsed, loss, accuracy, and learning rate, used by proposing algorithm, respectively. Training was done on a single GPU with initializing flower image normalization by 64*64*3.

| Epoch | Iteration | Time Elapsed (hh:mm:ss) | Mini-batch Loss | Mini-batch Accuracy | Base-Learning Rate |
|-------|-----------|--------------------------|-----------------|---------------------|--------------------|
| 1     | 1         | 00:00:00                 | 10.00%          | 9.9191              | 1.0000e-04         |
| 1     | 50        | 00:00:02                 | 40.00%          | 1.7673              | 1.0000e-04         |
| 1     | 100       | 00:00:03                 | 30.00%          | 0.8035              | 1.0000e-04         |
| 1     | 150       | 00:00:04                 | 40.00%          | 1.3736              | 1.0000e-04         |
| 2     | 200       | 00:00:05                 | 50.00%          | 1.2381              | 1.0000e-04         |
| 2     | 250       | 00:00:06                 | 60.00%          | 0.8936              | 1.0000e-04         |
| 2     | 300       | 00:00:07                 | 40.00%          | 1.5919              | 1.0000e-04         |
| 3     | 350       | 00:00:07                 | 70.00%          | 1.6425              | 1.0000e-04         |
| 3     | 400       | 00:00:08                 | 70.00%          | 0.7327              | 1.0000e-04         |
| 3     | 450       | 00:00:09                 | 40.00%          | 1.1009              | 1.0000e-04         |
| 4     | 500       | 00:00:10                 | 70.00%          | 1.3039              | 1.0000e-04         |
| 4     | 550       | 00:00:11                 | 90.00%          | 0.4999              | 1.0000e-04         |
| 4     | 600       | 00:00:12                 | 40.00%          | 0.9247              | 1.0000e-04         |
| 5     | 650       | 00:00:13                 | 70.00%          | 1.3738              | 1.0000e-04         |
| 5     | 700       | 00:00:14                 | 100.00%         | 0.2903              | 1.0000e-04         |
| 5     | 750       | 00:00:15                 | 50.00%          | 0.8617              | 1.0000e-04         |

The proposed algorithm experimented with different number of flower images and maximum epoch was 20. From the results of proposed method, it is clear that when NOI = 1650 and epoch is equal to 20, the time required for training is 55s and accuracy is 98% with minimum batch loss equal to 0.4728 and learning rate is 1.00e-04.
|   |   |     |     |      |      |
|---|---|-----|-----|------|------|
| 6 | 800| 00:00:16| 80.00%| 1.3198| 1.0000e-04 |
| 6 | 850| 00:00:17| 100.00%| 0.4936| 1.0000e-04 |
| 6 | 900| 00:00:18| 60.00%| 0.8758| 1.0000e-04 |
| 7 | 950| 00:00:19| 100.00%| 1.2795| 1.0000e-04 |
| 7 | 1000| 00:00:20| 100.00%| 0.3205| 1.0000e-04 |
| 7 | 1050| 00:00:21| 90.00%| 0.8462| 1.0000e-04 |
| 8 | 1100| 00:00:22| 100.00%| 0.4936| 1.0000e-04 |
| 8 | 1150| 00:00:23| 90.00%| 0.3343| 1.0000e-04 |
| 8 | 1200| 00:00:24| 90.00%| 0.8591| 1.0000e-04 |
| 9 | 1250| 00:00:24| 100.00%| 1.2795| 1.0000e-04 |
| 9 | 1300| 00:00:25| 90.00%| 0.2876| 1.0000e-04 |
| 9 | 1350| 00:00:26| 100.00%| 0.7411| 1.0000e-04 |
| 10 | 1400| 00:00:27| 100.00%| 1.1223| 1.0000e-04 |
| 10 | 1450| 00:00:28| 60.00%| 0.2202| 1.0000e-04 |
| 10 | 1500| 00:00:29| 60.00%| 0.9262| 1.0000e-04 |
| 11 | 1550| 00:00:30| 100.00%| 1.2601| 1.0000e-04 |
| 11 | 1600| 00:00:31| 80.00%| 0.2284| 1.0000e-04 |
| 11 | 1650| 00:00:32| 40.00%| 0.7777| 1.0000e-04 |
| 12 | 1700| 00:00:33| 100.00%| 1.6023| 1.0000e-04 |
| 12 | 1750| 00:00:33| 60.00%| 0.2492| 1.0000e-04 |
| 12 | 1800| 00:00:34| 80.00%| 0.6705| 1.0000e-04 |
| 13 | 1850| 00:00:35| 80.00%| 1.2013| 1.0000e-04 |
| 13 | 1900| 00:00:36| 60.00%| 0.1893| 1.0000e-04 |
| 13 | 1950| 00:00:37| 80.00%| 0.7223| 1.0000e-04 |
| 14 | 2000| 00:00:38| 100.00%| 1.1131| 1.0000e-04 |
| 14 | 2050| 00:00:39| 40.00%| 0.2375| 1.0000e-04 |
| 14 | 2100| 00:00:40| 60.00%| 0.7063| 1.0000e-04 |
| 15 | 2150| 00:00:41| 100.00%| 1.1984| 1.0000e-04 |
| 15 | 2200| 00:00:41| 60.00%| 0.1925| 1.0000e-04 |
| 15 | 2250| 00:00:42| 60.00%| 0.6344| 1.0000e-04 |
| 16 | 2300| 00:00:43| 100.00%| 1.1999| 1.0000e-04 |
| 16 | 2350| 00:00:44| 80.00%| 0.1033| 1.0000e-04 |
| 16 | 2400| 00:00:45| 80.00%| 0.5931| 1.0000e-04 |
| 17 | 2450| 00:00:46| 100.00%| 1.1842| 1.0000e-04 |
| 17 | 2500| 00:00:47| 80.00%| 0.1606| 1.0000e-04 |
| 17 | 2550| 00:00:47| 80.00%| 0.5756| 1.0000e-04 |
| 18 | 2600| 00:00:48| 100.00%| 1.0810| 1.0000e-04 |
| 18 | 2650| 00:00:49| 80.00%| 0.0858| 1.0000e-04 |
| 18 | 2700| 00:00:50| 80.00%| 0.5487| 1.0000e-04 |
| 19 | 2750| 00:00:51| 100.00%| 1.0917| 1.0000e-04 |
| 19 | 2800| 00:00:52| 90.00%| 0.0808| 1.0000e-04 |
Columns in Table 5 shows the NOI (number of images of each type of flower and there are five types of flower images used by proposed algorithm for simulation), epoch, time elapsed, loss, accuracy, and learning rate, used by proposed algorithm, respectively. The training was done on a single GPU with initializing flower image normalization by 64*64*3.

The proposed algorithm experimented with different number of flower image and maximum epoch was 20. From the results of proposed method, it is clear that when NOI = 165 and epoch is equal to 20, then the time required for training is 00.55s and accuracy up to be 98% with minimum batch loss equal to 0.4728 and learning rate is 1.00e-04. The figure 14 gives a Training progress graph of the Accuracy and Loss with number of iteration over the proposed algorithm when run on GPU with NOI = 165 of each class (number of images).

Table 6 shows the flower image detection time taken by proposed algorithm. When the proposed algorithm is executed on a high computation setup, it has an approximately 98% accuracy. This result compared with experiment_1 and it seems that the time required to detect the flower image by the proposed system is less when the same method runs on a high computation setup.
TABLE VI
FLOWER CLASSIFICATION RESULTS WITH TIME

| Types of Flower | Time required for Flower classifier (GPU) | Time required for Flower classifier (CPU) |
|-----------------|----------------------------------------|----------------------------------------|
|                 | Experiment_2                           | Experiment_1                           |
| Tulip           | 9.987s                                 | 16.183s                                |
| Rose            | 9.591s                                 | 13.989s                                |
| Dandelion       | 10.182s                                | 10.182s                                |
| Daisy           | 5.788s                                 | 10.101s                                |

V. CONCLUSION
An extended hybrid methodology based on deep learning CNN is proposed for the detection and segmentation of flower RGB images with L*a*b* conversion. The experimental results show that the proposed methodology gives better accuracy and minimizes the error rate for RGB flower image segmentation.

- Experimental outcomes show that the accuracy of deep CNN depends on the weight function on the different layers if the value of weight is changed the learning accuracy is affected by the proposed method.
- The learning accuracy of deep convolution neural networks also depends on the Learning Rates, Batch Loss value, and types of images used for training deep convolution neural network.
- The accuracy is also affected by hardware configuration used by the implementation purpose.

The proposed deep learning-based convolution neural network provides the best flower classification results on all the datasets. There are various convolution neural network architectures like LeNet, AlexNet, GoogLeNet, VGGNet used to solve deep learning computer vision problems. In a future enhancement, the proposed system would be compared with a pertained network like LeNet, AlexNet, and GoogLeNet. The proposed system is applicable for Image Classification, Object Recognition, Content-Based Image retrieval system, Self-driving cars, Speech Recognition, etc.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent to publish
Not applicable.

Availability of data and materials
The datasets generated during and/or analyzed during the current study are not publicly available, but are available from the corresponding author who was an organizer of the study.

Competing interests
The authors declare that they have no competing interests.

Funding
No Funding till date but we applied for in a government organization.

Authors' contributions
All of the authors contributed to the conception of the study, drafting and critical revision of the manuscript, and provided final approval of the manuscript.

REFERENCES

[1] Dhanalakshmi, S., and Dr T. Ravichandran. "A new method for image segmentation." International Journal of Advanced Research in Computer Science and Software Engineering 2, no. 9 (2012): 293-299.

[2] Tan, Y. J., K. S. Sim, and F. F. Ting. "Breast cancer detection using convolutional neural networks for mammogram imaging system." In 2017 International Conference on Robotics, Automation and Sciences (ICORAS), pp. 1-5. IEEE, 2017.

[3] Baziyad M, Rabie T, Kamel I. Extending steganography payload capacity using the L*a*b* color space. In 2018 International Conference on Innovations in Information Technology (IIT) 2018 Nov 18 (pp. 1-6). IEEE.

[4] Baziyad, Mohammed, Tamer Rabie, and Ibrahim Kamel. "Extending steganography payload capacity using the L*a*b* color space." In 2018 International Conference on Innovations in Information Technology (IIT), pp. 1-6. IEEE, 2018.

[5] Chen, Yang, Won Suk Lee, Hao Gan, Natalia Peres, Clyde Fraise, Yanchao Zhang, and Yong He. "Strawberry yield prediction based on a deep neural network using high-resolution aerial orthoimagery." Remote Sensing 11, no. 13 (2019): 1584.

[6] Hiary, Hazem, Heba Saadeh, Maha Saadeh, and Mohammad Yaqub. "Flower classification using deep convolutional neural networks." IET Computer Vision 12, no. 6 (2018): 855-862.

[7] Gavai, Nitin R., Yashashree A. Jakhade, Seema A. Tribhuvan, and Rashmi Bhattad. "MobileNets for flower classification using TensorFlow." In 2017 International Conference on Big Data, IoT and Data Science (BID), pp. 154-158. IEEE, 2017.

[8] Xia, Xiaodong, Cui Xu, and Bing Nan. "Inception-v3 for flower classification." In 2017 2nd International Conference on Image, Vision and Computing (ICIVC), pp. 783-787. IEEE, 2017.

[9] Gogul, I., and V. Sathikesh Kumar. "Flower species recognition system using convolution neural networks and transfer learning." In 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), pp. 1-6. IEEE, 2017.

[10] Lodh, Avishikta, and Ranjan Parekh. "Flower recognition system based on color and GIST features." In 2017 Devices for Integrated Circuit (DevIC), pp. 790-794. IEEE, 2017.

[11] Grauman, Kristen, and Trevor Darrell. "The pyramidal match kernel: Discriminative classification with sets of image features." In Tenth IEEE International Conference on Computer Vision (ICCV05) Volume 1, vol. 2, pp. 1458-1465. IEEE, 2005.

[12] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431-3440. 2015.

[13] Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580-587. 2014.

[14] Gromenspan, Hayit, Bram Van Ginneken, and Ronald M. Summers. "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique." IEEE Transactions on Medical Imaging 35, no. 5 (2016): 1153-1159.

[15] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In Advances in neural information processing systems, pp. 1097-1105. 2012.

[16] Dou, Qi, Hao Chen, Lequan Yu, Lei Zhao, Jing Qin, Defeng Wang, Vincent CT Mok, Lin Shi, and Pheng-Ann Heng. "Automatic detection of cerebral microbleeds from MR images via 3D convolutional neural networks." IEEE transactions on medical imaging 35, no. 5 (2016): 1182-1195.

[17] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431-3440. 2015.

[18] Li, Boxun, Yuzhi Wang, Yu Wang, Yiran Chen, and Huazhong Yang. "Training itself: Mixed-signal training acceleration for memristor-based neural network." In 2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 361-366. IEEE, 2014.

[19] Ioffe, Sergey, and Christian Szegedy. "Batch normalization:
Accelerating deep network training by reducing internal covariate shift." \textit{arXiv preprint arXiv:1502.03167} (2015).

[20] Donahue, Jeff, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. "DeepLab: A deep convolutional activation feature for generic visual recognition." In \textit{International conference on machine learning}, pp. 647-655. 2014.

[21] Zhang L, Lu L, Nogues I, Summers RM, Liu S, Yao J. DeepPap: deep convolutional networks for cervical cell classification. IEEE journal of biomedical and health informatics. 2017 May 19;21(6):1633-43.

[22] Zhang, Ling, Le Lu, Isabella Nogues, Ronald M. Summers, Shaoxiong Liu, and Jianhua Yao. "DeepPap: deep convolutional networks for cervical cell classification." \textit{IEEE journal of biomedical and health informatics} 21, no. 6 (2017): 1633-1643.

[23] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." \textit{arXiv preprint arXiv:1409.1556} (2014).

[24] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." In \textit{International Conference on Medical image computing and computer-assisted intervention}, pp. 234-241. Springer, Cham, 2015.

[25] Zhu, Xiaofeng, Heung-II Suk, Li Wang, Seong-Whan Lee, Dinggang Shen, and Alzheimer’s Disease Neuroimaging Initiative. "A novel relational regularization feature selection method for joint regression and classification in AD diagnosis." \textit{Medical image analysis} 38 (2017): 205-214.

[26] Maje, Subhransu, Alexander C. Berg, and Jitendra Malik. "Classification using intersection kernel support vector machines is efficient." In \textit{2008 IEEE conference on computer vision and pattern recognition}, pp. 1-8. IEEE, 2008.

[27] Ito, Satoshi, and Susumu Kubota. "Object classification using heterogeneous co-occurrence features." In \textit{European Conference on Computer Vision}, pp. 701-714. Springer, Berlin, Heidelberg, 2010.

[28] Liu, Zeyu, Yantaoy Yang, and Qingdong Cai. "Neural network as a function approximator and its application in solving differential equations." \textit{Applied Mathematics and Mechanics} 40, no. 2 (2019): 237-248.

[29] Donahue, Jeff, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. "Decaf: A deep convolutional activation feature for generic visual recognition." In \textit{International conference on machine learning}, pp. 647-655. 2014.

[30] Jaisswal, Varshali, Varsha Sharma, and Sunita Varma. "MMFO: modified moth flame optimization algorithm for region based RGB color image segmentation." \textit{International Journal of Electrical & Computer Engineering} (2008-8708) 10 (2020).

[31] Rathore, Neeraj, and Indeverve Chana. "Load balancing and job migration techniques in grid: a survey of recent trends." \textit{Wireless personal communications} 79, no. 3 (2014): 2089-2125.

[32] Rathore, Neeraj Kumar. "Ethical hacking & security against cyber crime." \textit{Journal on Information Technology (JIT)} 5, no. 1 (2016): 7-11.

[33] Rathore, Neeraj, and Indeverve Chana. "Job migration with fault tolerance based QoS scheduling using hash table functionality in social Grid computing." \textit{Journal of Intelligent & Fuzzy Systems} 27, no. 6 (2014): 2821-2833.

[34] Rathore NK, and Chana "Comparative analysis of checkpointing", \textit{In PIMR 3rd National IT conference, IT enabled practices and emerging management Paradigm book and category is

communication technologies and security issues, Topic NoName-46, Prestige Management and Research, Indore}, pp 32-35. 2008

[35] Rathore, Neeraj Kumar, and Indeverve Chana. "A cognitive analysis of load balancing technique with job migration in grid environment." In \textit{World congress on information and communication technology (WICT), Mumbai}, IEEE proceedings paper, pp. 77-82. 2011.

[36] Rathore, Neeraj, and Indeverve Chana. "Variable threshold-based hierarchical load balancing technique in Grid." \textit{Engineering with computers} 31, no. 3 (2015): 597-615.

[37] Rathore, Neeraj, and Indeverve Chana. "A sender initiate based hierarchical load balancing technique for grid using variable threshold value." In \textit{2013 IEEE International Conference on Signal Processing, Computing and Control (ISPCC)}, pp. 1-6. IEEE, 2013.

[38] Rathore, Neeraj Kumar. "Map reduce architecture for grid." \textit{i-manager's Journal on Software Engineering} 10, no. 1 (2015): 21-30.

[39] Rathore, Neeraj Kumar. "Efficient hierarchical load balancing technique based on grid." In \textit{29th MP Young scientist congress, Bhopal, MP}, p. 55. 2014.

[40] Rathore, Neeraj. "Performance of hybrid load balancing algorithm in distributed web server system." \textit{Wireless Personal Communications} 101, no. 3 (2018): 1233-1246.

[41] Rathore, Neeraj. "Efficient Agent Based Priority Scheduling and Load Balancing Using Fuzzy Logic in Grid Computing, i-manager's Journal on Computer Science, 3 (3) Sep-Nov 2015 Print ISSN 2347-2227 (2015): 2347-6141.

[42] Jain, Neelesh Kumar, Neeraj Kumar Rathore, and Amit Mishra. "An efficient image forgery detection using biorthogonal wavelet transform and improved relevance vector machine." \textit{Wireless Personal Communications} 101, no. 4 (2018): 1983-2008.

[43] Rathore, Neeraj. "Dynamic threshold based load balancing algorithms." \textit{Wireless Personal Communications} 91, no. 1 (2016): 151-185.

[44] Rathore, Neeraj Kumar, Umashankar Rawat, and Satish Chandra Kulhari. "Efficient hybrid load balancing algorithm." \textit{National Academy Science Letters} 43, no. 2 (2020): 177-185.

[45] Jain, Neelesh Kumar, Neeraj Kumar Rathore, and Amit Mishra. "An efficient image forgery detection using biorthogonal wavelet transform and improved relevance vector machine." \textit{Wireless Personal Communications} 101, no. 4 (2018): 1983-2008.

[46] Rathore, Neeraj Kumar, and Indeverve Chana. "Job migration policies for grid environment." \textit{Wireless Personal Communications} 89, no. 1 (2016): 241-269.

[47] Sharma, Vishal, Rajesh Kumar, and Neeraj Rathore. "Topological Broadcasting Using Parameter Sensitivity-Based Logical Proximity Graphs in Coordinated Ground-Flying Ad Hoc Networks." \textit{J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.}, 6, no. 3 (2015): 54-72.

[48] Rathore, Neeraj, and Indeverve Chana. "Report on hierarchical load balancing technique in grid environment." \textit{i-manager's Journal on Information Technology 2}, no. 4 (2013): 21.