IET Research Journals

IET Journals
The Institution of Engineering and Technology

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

Flower Classification using Deep Convolutional Neural Networks

Hazem Hiary[1]*, *Heba Saadeh*[1], *Maha Saadeh*[1], *Mohammad Yaqub*[2]

[1] *Computer Science Department, The University of Jordan, Amman, Jordan*
[2] *Institute of Biomedical Engineering, Department of Engineering Science, University of Oxford, Oxford, UK*
* *E-mail: hazemh@ju.edu.jo*

Abstract: Flower classification is a challenging task due to the wide range of flower species which have similar shape, appearance or surrounding objects such as leaves and grass. In this paper, we propose a novel two-step deep learning classifier to distinguish flowers of a wide range of species. Firstly, the flower region is automatically segmented to allow localisation of the minimum bounding box around it. The proposed flower segmentation approach is modelled as a binary classifier in a fully convolutional network framework. Secondly, we build a robust convolutional neural network classifier to distinguish the different flower types. We propose novel steps during the training stage to ensure robust, accurate and real-time classification. We evaluate our method on three well known flower datasets. Our classification results exceed 97% on all datasets which is better than the state-of-the-art in this domain.

1 Introduction

Unlike simple object classification such as distinguishing cats from dogs, flower recognition and classification is a challenging task due to the wide range of flower classes that share similar features: Several flowers from different types share similar colour, shape and appearance. Furthermore, images of different flowers usually contain similar surrounding objects such as leaves, grass, etc. There are more than 250000 known species of flowering plants classified into about 350 families [1]. A wide range of various applications including content-based image retrieval for flower representation and indexing [2], plants monitoring systems, floriculture industry [3], live plant identification and educational resources on flower taxonomy [4] depend on successful flower classification. Manual classification is possible but time consuming and tedious to use with a large number of images and potentially erroneous in some flower classes especially when the image background is complex. Thus, robust techniques of flower segmentation, detection and classification have great value.

Conventional flower classification techniques use a combination of features extracted from the flower images with the aim of improving classification performance [5–7]. Colour, texture, shape, and some statistical information are among the main sources of features that are widely used to identify the different flower species [5, 8–10]. Some methods rely on human interaction to further enhance the classification results [7, 11, 12]. In addition, Support Vector Machines (SVM) are among the most commonly used types of classifiers [5, 13, 14]. Many flower classification techniques rely on learning their features from a segmented flower region to improve accuracy [5, 15–17].

Hand-crafted traditional discriminative features that can be used in a classification task such as histogram of oriented gradients (HOG), scale-invariant feature transform (SIFT), speeded up robust features (SURF), etc., cannot be easily applied to the flower classification problem due to the problem complexity as well as the numerous flower classes. In addition, robustness of a flower classification technique applied to one flower dataset is not guaranteed on a different flower dataset. This is mainly because conventional methods rely heavily on specific hand-made features which might not be generalisable to other flower images or similar flower images with different conditions such as change of lightning, flower pose or variation of surrounding objects.

Deep learning techniques, especially Convolutional Neural Networks (CNNs), have recently gained wide interest due to superior accuracy compared to classical machine learning methods which rely on hand-crafted features. In addition, the advance of hardware capabilities particularly with the use of Graphics Processing Units (GPUs) sped up the processing time of deep learning techniques significantly [18, 19].

In this work, we show how we utilise recent development of deep learning methods such as CNN alongside the existence of reasonable size flower datasets to tackle the flower classification task robustly. Our automatic method detects the region around the flower in an image, then uses the cropped images to learn a strong CNN classifier to distinguish different flower classes. The detection is performed by finding the minimum bounding box around an automatically segmented flower. The segmentation is achieved as a binary classification task within a Fully Convolutional Network (FCN) [20] framework. Our robust method is evaluated on different known flower datasets and results show that the proposed technique achieves at least 97% classification accuracy on all datasets.

The rest of this paper is organised as follows, in Section 2 we present the background and related work. Section 3 presents the proposed method. The experimental setup is described in Section 4, followed by results and comparisons in Section 5. We then conclude our work in Section 6.

2 Related work

In this section, we describe CNN and its application in image classification and segmentation. We then present related work which addresses the flower segmentation and classification task. We generally split the techniques to deep learning and non-deep learning based techniques.

2.1 CNN for image classification and segmentation

A Convolutional neural network consists of a number of convolutional and subsampling layers optionally followed by fully connected layers. For the sake of this work, we focus on 2D CNNs which typically work on 2D images; although 1D or higher dimensional CNNs have similar concepts.

The input to a convolutional layer is an $(r \times c \times n)$ image I where r is the number of rows, c is the number of columns and n is the

number of channels. The convolutional layer is meant to learn K filters (or kernels) of size $(k_r \times k_c \times k_n)$.

In addition, padding the input image with p (p_r, p_c) pixels permits convolution for pixels at the border of the image. p is typically set as half of the kernel size $(k_r/2, k_c/2)$. Furthermore, a stride value s defines the kernel movement over the image. After the convolution of an input image with K kernels, the resultant K feature maps have the following size

$$M_r^k = \frac{r - k_r + 2 \times p_r}{s_r} + 1 \quad (1)$$

where M_r^k is the number of rows in the k^{th} feature map. The number of columns in a feature map M_c^k is similarly derived. A non-linearity transformation, e.g., Rectified Linear Unit (ReLU), is typically applied to all feature maps after the convolutional layer to speed up the training process [18]. Moreover, each feature map is then down-sampled typically with a pooling step which reduces the size of the feature maps and allows the next convolutional layer to work on a larger receptive field compared to the first one. This helps the network learn features at multiple scales. The generated feature maps after the first convolutional, non-linearity and pooling layers are then passed as input to the next block of layers to compute the next set of feature maps and so on. Optionally, fully connected layers can be used at the end of the CNN to determine which features most correlate to a particular class. The output of the last layer is an N-dimensional vector where N is the number of classes in a given problem.

During the training stage, a loss function, such as the mean square error in Eq. (2), is used to compute the difference between the actual (y) and predicted $(\tilde{y})$ labels.

$$err = \frac{1}{2} \sum (y - \tilde{y})^2 \quad (2)$$

The use of the CNN is expanded to allow image segmentation and object detection. Image segmentation using CNN can be performed using a concept called Fully Convolutional Network (FCN) for semantic segmentation [20]. In addition, methods have been proposed to allow the CNN to do object detection such as region proposals with CNN (R-CNN) [21], fast R-CNN [22], faster R-CNN [23] and YOLO [24]. Overall, these techniques and the FCN method provide similar results on benchmarked models such as AlexNet [18] and VGG-16 [19]. In this paper, we focus on the FCN model to segment then detect the flower region mainly because it can be easily reused in the classification model as described in Section 3.

FCN can be considered as a special type of CNN in which deconvolutional layer(s) can be used to up-sample and fuse the feature maps from some convolutional layer(s) such that a segmentation mask can be learnt. Typically, the segmentation mask is the same size of the input image which provides a pixel-wise classification for each pixel.

2.2 Flower segmentation and classification

Various approaches have been proposed to classify flower images. The majority of researchers have used machine learning based methods. For instance, the work in [5] segmented and classified flowers using SVM and multiple kernel learning (MKL). They extracted features from SIFT, HOG, and the HSV colour model. This work has later been improved in [15] and then further advanced in [13]. In [15] they used the concept of BiCoS (Bi-level Co-Segmentation), and BiCoS-MT (multi-task) in an SVM classifier while in [13] they used TriCoS (Tri-level Co-segmentation) to tackle the flower segmentation and classification; an SVM model was used with SIFT, Lab colour model, principle component analysis (PCA), fisher vector (FV), and Gaussian mixture model (GMM).

A user-interactive method computer assisted visual interactive recognition (CAVIAR) was proposed in [7] which extracts shape features from a rose curve model, and hue and saturation colour moments. A classification approach is proposed in [11] using weighted Euclidean distance with features from the HSV colour model and boundary shape of flowers. They also extracted colour and shape features from flower centre area. However, this method requires manual user interaction.

Other approaches in flower classification have been proposed, such as pairwise rotation invariant co-occurrence local binary pattern (PRICoLBP) [16]; metric forests with GMM [25]; generalised max pooling (GMP) with FV and power normalisation [26]; visual adjectives (VA) with SIFT and improved FV [27]; saliency driven image multiscale nonlinear diffusion filtering [17]; heterogeneous co-occurrence features [28]; generalised hierarchical matching (GHM) with saliency map (LocSaliency) [14]; contextual exemplar classifier (CEC) [29]; fisher discrimination dictionary learning (FDDL) with frequent local histograms (FLH) [8]; grid-specific bag-of-FLH (GRID-FLH) [30]; colour attention-based bag-of-words [9]; Harr-like transformation of local features [31]; and graph-regularised robust late fusion (GRLF) [32]. All the aforementioned methods rely on hand-crafted features which use classical classifiers such as SVM.

On the other hand, deep learning techniques, especially CNN-based, were proposed to tackle the flower classification task. CNNs have recently gained a lot of interest in solving several learning problems due to superior accuracy compared to classical methods. They have been recently used in several natural image classification tasks [18, 19, 33, 34].

There are a handful of works in the literature which use CNN to address the flower classification problem [10, 35–48]. For instance, the work in [35] approached the problem using a two-level hierarchical feature leaning that used a deep CNN. They first used a transfer learning method (hierarchical feature learning (HFL)) to initialise a pre-trained deep CNN model for the new target dataset. The deep feature extractors at different levels were then trained. This method effectively increases the classification accuracy in comparison with other classification methods.

A combined online nearest-neighbour estimation (ONE) algorithm was proposed for both image classification and retrieval [36]. Manual object definition, regional description and nearest-neighbour search of extracted CNN features were involved in this algorithm by computing similarity between the query and each category or image candidate. Results show state-of-the-art accuracy in a wide range of image classification and retrieval datasets with reasonable computational overheads. The work in [37] addressed different recognition tasks including flower classification using standard CNN representation called OverFeat. The experimental study shows significant results in the different classification tasks on various datasets.

The authors in [10], on the other hand, proposed reversal-invariant deep features (RI-Deep), and reversal invariant convolution (RI-Conv) layers to increase the CNN capacity without affecting the model complexity. On various image classification tasks, this approach shows an improvement in classification accuracy, including scene understanding, fine-grained object recognition, and large-scale visual recognition. The authors in [38] proposed an approach to extract deep convolutional activation features (DeCAF) to use with a k-NN classifier. Their empirical study shows that the proposed method can yield an improved accuracy and performance compared to state-of-the-art approaches.

A task-driven pooling (TDP) model to learn pooled representation implicitly from data was presented in [39]. TDP was used to replace average or max pooling in CNN models to achieve a better pooled representation. The proposed method was extended to multitask classification to maximise the accuracy on a flower dataset. In different work, guidelines on how to properly transfer CNN features to solve a specific task were discussed in [40]. Their evaluation showed state-of-the-art improvement on different datasets including a flower dataset.

Recently, a method to speed up the computational time of the CNN forward and backward propagation steps using winner takes all (WTA) hashing was described in [41]. In a different approach, a hierarchical deep semantic representation (H-DSR) which combines semantic context modelling with visual features was proposed in [42]. Deep CNN features were extracted from spatially fixed image grids to detect a response map using pre-learned classifiers. The response map was then used to extract semantic representation,

which is further combined with visual representations to form a hierarchical deep semantic model. In work related to ours, a CNN-based method to perform flower classification was proposed in [43]. They used luminance and saliency map approaches to select the flower region. The method was evaluated on a flower dataset.

A convolutional fusion networks (CFN) model to fuse multi-scale deep representations was proposed in [44]. This model adds more parameters to generate new side branches from the intermediate layers, and learns adaptive weights for these branches. However, the accuracy reported on flower classification is limited compared to our work and other published work such as [40]. Authors in [45] proposed a collaborative representation based classification (CRC) approach, which represents the image as a weighted collaboration of features over all classes. They extracted features using different descriptors including CNN-based features and used these features in the classification task.

An approach based on the CNN Inception model was proposed in [46] for flower classification. The method was applied to the Oxford 17 and Oxford 102 datasets and achieved good results. A generic approach based on unsupervised fine-grained image retrieval in different applications including flowers was proposed in [47]. No annotation was needed to cluster the objects as the proposed method relies on detecting the main object in an image to create deep descriptors for image categorisation. Finally, a fine-grained recognition approach based on local parts and global discrimination CNN was proposed in [48]. The method was applied to different sets including Oxford 102. The proposed CNN consists of two networks with shared weights such that one network is focused on the local parts of the input image while the second on the global geometry of the image.

3 Proposed method

We propose a two-step approach for the flower classification problem. The first step localises the flower by detecting the minimum bounding box around it. The localisation is performed by segmenting the flower region using an FCN method [20]. The second step learns a CNN to accurately classify the different flower classes. Figure 1 shows the overall framework for the proposed method. Here we show how the segmentation FCN is initialised by the VGG-16 model [19] while the classification CNN is initialised by segmentation FCN.

3.1 Network initialisation via transfer-learned ImageNet features

Although kernels in the convolutional networks can be initialised randomly, most deep learning methods utilise the existence of pre-trained models on large datasets such as ImageNet for initialising, i.e., transfer-learning, their models. This helps train networks for problems with small numbers of training examples since many image classification applications share similar low level features e.g., edges, blobs, etc.

We initialise the proposed FCN from the VGG-16 model [19] which provided robust results on classifying images from the ImageNet dataset. The trained FCN is then used to initialise the classification CNN. The VGG-16 model consists of 5 convolutional blocks followed by 3 fully connected layers. Each convolutional block consists of 2 or 3 convolutional layers and ReLU. At the end of each convolutional block a max pooling layer is used to down-sample the feature maps which makes the features translation and scale invariant. Figure 2 shows a detailed description of the VGG-16 model alongside the parameters of the proposed FCN.

Although there are new published models such as [33, 34] that exceeded the VGG-16 ImageNet classification accuracy, we have chosen this model to initialise our FCN and consequently the CNN model because it better suits the flower classification task. Deeper models such as resNet [33] are generally too complex to handle this task because the number of parameters is a over-kill. In fact, we show here how we initialise our models by a reduced version of the VGG-16 model with no compromise on accuracy.

3.2 Fully convolutional network for semantic flower segmentation

Flower images usually contain wide surrounding clutter which make the problem of automatic flower classification challenging. Therefore, we propose an automatic step which allows the detection of the flower within the image by segmenting the flower region only using FCN [20]. We formulate the segmentation task as a binary classification problem; i.e., 0 for background and 1 for the flower region(s).

Our proposed FCN consists of several convolutional layers and three de-convolutional layers. The network is initialised by the first 5 blocks from the VGG-16 model (Fig. 2). Each convolutional layer learns K kernels and produces K feature maps by sliding each 2D kernel over the input image from the previous layer [49]. The 2D feature map value at position (i, j) is computed as

$$f_{i,j} = \sum_{a=-k_r/2}^{k_r/2} \left(\sum_{b=-k_c/2}^{k_c/2} k_{a,b} \times I_{i+a,j+b} \right) \quad (3)$$

The 3 de-convolutional layers up-sample the feature maps generated from the fifth block gradually. The first and second de-convolutional layers use a stride of 2 while the final de-convolutional layer uses a stride of 8. This is usually referred to as FCN-8s [20]. We use bi-linear interpolation up-sampling to ensure smooth reconstruction of edges of the flower. In addition, the (2, 2, 8) up-sampling also allows fine-grained interpolation which permits the segmentation of fine detailed structures. Notice that our model size is approximately 80% smaller than the VGG-16 model because we dropped the fully connected layers from the model, which are shown as blocks with dashed border in Fig. 2.

We use backpropagation to train the FCN. However, we initially fix the kernels on blocks 1 and 2 (i.e., use the ImageNet kernels) and only let the model learn the kernels on blocks 3, 4 and 5. This allows it to learn the mid-to-global feature maps without optimising the low level kernels. When the validation accuracy saturates, we stop training and then restart again starting from the last learned model to let the FCN learn the kernels in the first two blocks. This permits the model to learn local features. We found experimentally that this process improves segmentation accuracy compared to learning all kernels in one step. In addition, due to having a small dataset, we augment the images during training by allowing small rotation, horizontal flipping, random small cropping of the image border, or any combination of these transformation. This helps create a more robust FCN model and avoids overfitting.

During testing an unseen flower image, the output of the model is a mask of the same size, as shown in Fig. 3(b). Before the masked flower is fed to the next step, i.e., flower classification, we perform two pre-processing steps. First, we find the largest connected component in the segmentation mask, as in the white region in Fig. 3(b), to keep the largest segmented flower region. This is important only when multiple flowers exist in one image; keeping only one flower region is sufficient and possibly less confusing for the classification task. Second, we use the minimum bounding box around the largest connected component (red box in Fig. 3(c)) to crop the original flower image while keeping the objects near the flower (as in Fig. 3(d)). These objects are mostly leaves, and it turns out that keeping them in the cropped image provides discriminative features when training the flower classifier since they retain important context for the flower.

3.3 Convolutional neural networks for flower classification

After generating cropped flower images, the task is simplified since the highly discriminative regions are mainly kept while other possible misleading regions are removed. In this work, we address the flower classification problem as a multi-class convolutional neural network classification of N classes. The problem is simply formulated as a function F which predicts the class c of an image x such as $c = F(x)$.

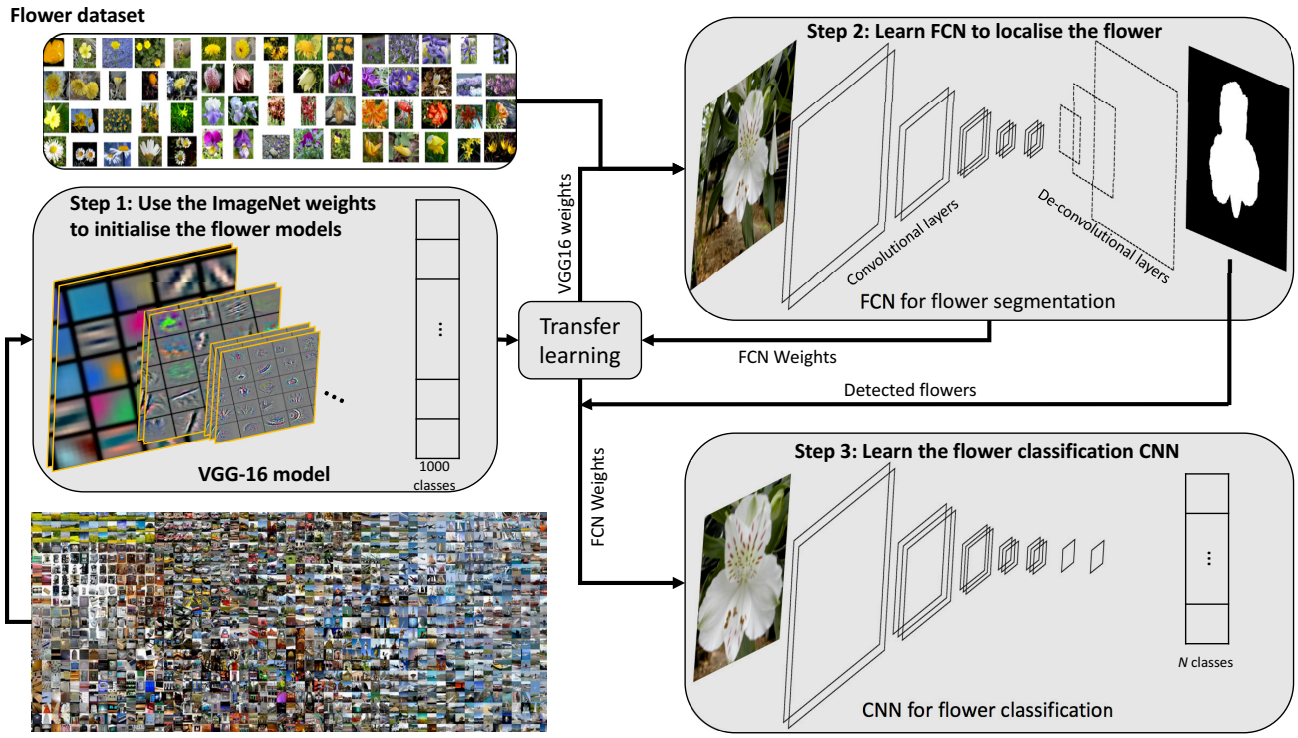

Fig. 1: Flow diagram of the proposed flower segmentation and classification method.

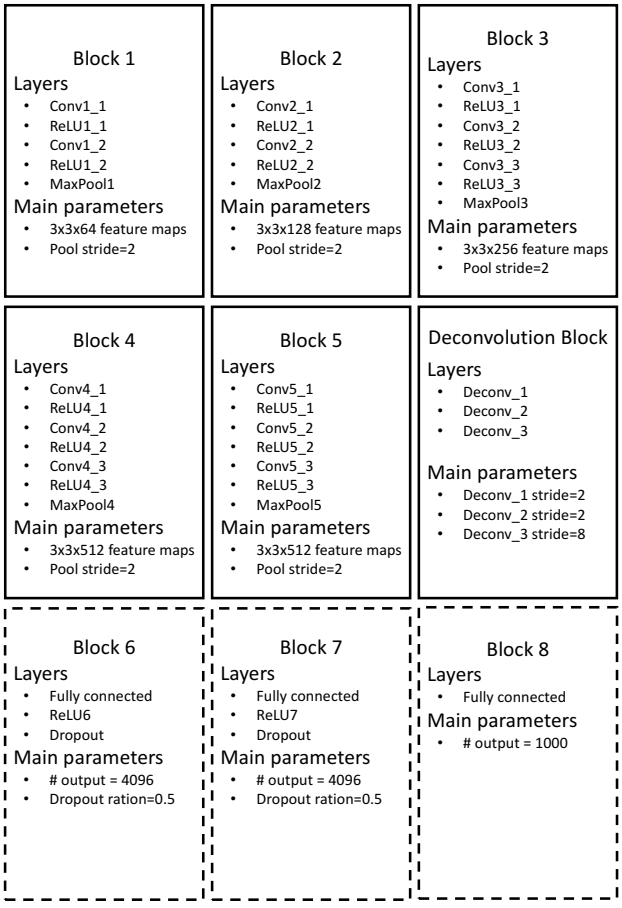

Fig. 2: The proposed FCN model and its detailed parameters shown in the boxes with solid border. The blocks with dashed border show the VGG-16 blocks which we excluded.

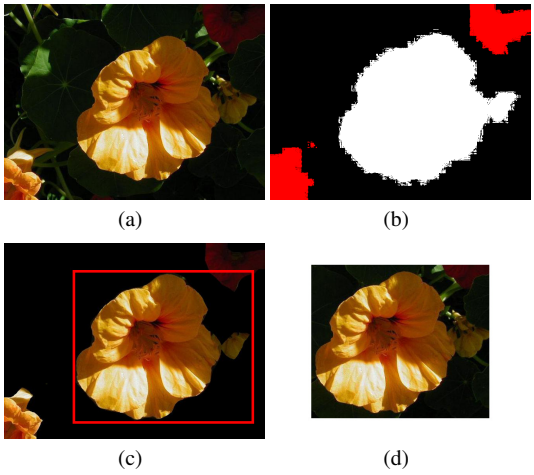

Fig. 3: FCN for flower segmentation, (a) original image, (b) mask of automatic segmentation (white region is the largest connected component and red regions are the small regions which are ignored during cropping the images), (c) the masked image that shows the minimum bounding box around the largest segmented flower region, (d) the cropped image region which is used in the classification step.

We propose a CNN which initialises its first 5 blocks from the FCN model which was already initialised by the VGG-16 model. However, instead of using 3 fully connected layers in blocks 6-8 (recall Fig. 2), we use 3 convolutional layers with 512 feature maps. The kernel size of the convolutional layer in block 6 is 7×7 while the number of output parameters from the convolutional layer in block 8 is N.

We use a multi-class Softmax loss function as a measure of the quality of a particular set of parameters based on how well the predicted outcomes match the ground truth labels in the training data. Softmax computes the probabilistic distribution over N different possible outcomes. We also use stochastic gradient descent (SGD) to optimise and update the set of parameters aiming to minimise the loss function. SGD and Softmax loss are commonly used

in other CNN-based applications such as [18, 19, 33, 34]. Our loss function takes as input an N-dimensional vector X and outputs an N-dimensional vector Y of real values between 0 and 1. This function is a normalised exponential and is defined as

$$\sigma(X)_j = \frac{e^{X_j}}{\sum_{n=1}^{N} e^{X_n}} \tag{4}$$

where $j = 1,\ldots,N$. We have noticed that the loss function was not performing well initially during CNN parameter optimisation. There are several possible reasons for this, but it is mainly due to the complexity of the multi-class classification problem compared to the learnt weights in the binary segmentation FCN. In addition, the function we are learning is not convex, not smooth and has many local minima with flat regions. Therefore, we propose two novel steps to improve the convergence of the algorithm.

Firstly, since we have more convolutional layers in the CNN than the FCN, we propose to learn kernel parameters in a 3-step approach. First, we let the CNN learn the kernels in the convolutional layers at blocks 6-8 while fixing the first 5 blocks. We then allow the CNN to learn the parameters in blocks 3-5. Finally, we let all parameters from all blocks be learned simultaneously. This provides better convergence for the CNN.

Secondly, because of the existence of a large number of flat local minima, the optimiser is prevented from reaching a good solution. Therefore, it is important to allow the optimiser in some scenarios to restart its search while finding a good minimum. To address this issue, we propose a multi-step training approach during which we force the learning rate to decrease in each step and then make a sudden large increase. The increase of the learning rate allows the optimiser to 'restart' itself to allow searching for other nearby solutions. More details about different approaches to restart SGD are described in [50].

In addition, thanks to the flower detection step described in Section 3.2, a wider range of augmentation can be used. For instance, a larger range of rotation angles and vertical flipping are used here than in the FCN model because the image is already cropped around the flower and large rotation does not affect the overall appearance of the whole cropped image. However, performing a large rotation on the whole (non-cropped) flower image may create a completely unrealistic image. Finally, with more possible augmentation, the generated CNN can be more robust to a wider range of transformations especially object rotation.

4 Experimental Setup

4.1 Datasets

Three datasets are used to test the proposed method; the Oxford 102 [5], the Oxford 17 [6], and Zou-Nagy [7]. Oxford 102 and Oxford 17 are two publicly available sets of flowers that have been widely used. The images have large scale, pose and light variations. The former set contains 8189 images from 102 flower categories with 40-258 images per category of various image size, while the latter consists of 1360 flower images from 17 categories, with 80 images in each category of various image size. Some flowers from the Oxford 17 are part of the Oxford 102. The third dataset which was compiled by Zou and Nagy consists of 612 flower images from 102 categories. Each category consists of six images and each image size is 300×240 pixels.

The variability of flower appearance, pose, zoom and surrounding objects is large in the Oxford images compared to Zou-Nagy. Flower images in the latter were consistently taken from a specific range of camera angle and distance. Therefore, this allows the flower images in this dataset to be more consistent and easier to distinguish (Fig. 4(a), 4(b) show some random but representative examples). On the other hand, Oxford flower images have greater variation as shown in Fig. 4(c), 4(d).

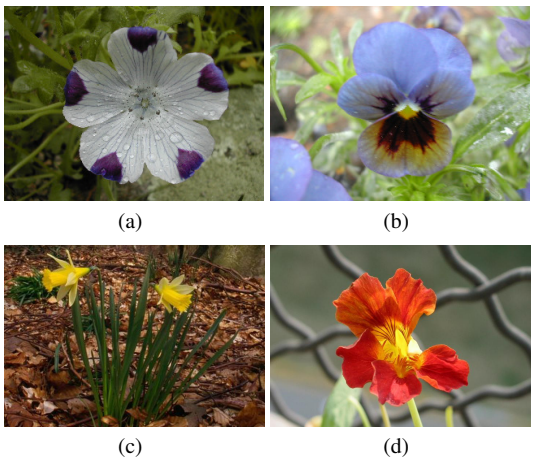

Fig. 4: Sample images from the three datasets to show the complex variability, (a) and (b) are from Zou-Nagy, (c) from Oxford 17 (cropped for better visualisation), (d) from Oxford 102.

4.2 Evaluation metrics

We propose several metrics to provide an insight into the accuracy of segmentation, detection and classification methods. Moreover, our accuracy measures allow us to do direct comparison with other published results.

For flower segmentation, we use pixel overlap score which is also known as intersection over union (IoU). IoU measures the percentage overlap of the intersected manual and automatic segmentation over their union (Eq. (5)). We only measure the overlap for the foreground object, i.e., the flower; and ignore background pixels. Overlap score has been used in some flower segmentation methods such as [51, 52]. The overlap value ranges from 0 to 1 such that the higher the value the more accurate the segmentation.

$$IoU = \frac{|\text{Manual} \cap \text{Automatic}|}{|\text{Manual} \cup \text{Automatic}|} \tag{5}$$

where $|\,.\,|$ is the cardinality of a set. Since our classification method is not sensitive to very accurate segmentation because it relies on a rough detection of the flower region, we evaluated the accuracy of flower detection. We propose a box overlap metric between the minimum bounding boxes around the manual and automatic segmentations. We compute the box IoU (B_{IoU}) which measures the box overlap between the manual and detected boxes. In addition, to decide the most acceptable threshold for B_{IoU}, i.e., the IoU threshold above which the two boxes are considered overlapped enough, we find $B^{th}_{overlap}$ (Eq. (6)) which computes the percentage of images having B_{IoU} greater or equal to an IoU threshold th. The value of th is varied between 0 (no overlap) to 1 (complete overlap).

$$B^{th}_{overlap} = \frac{|\text{images} : B_{IoU} \geq th|}{|\text{images}|} \tag{6}$$

Classification accuracy (acc) is measured as the number of correctly classified images over the total number of images such as

$$acc = \frac{|\text{images} : \text{predicted class} = \text{manual class}|}{|\text{images}|} \tag{7}$$

To understand the importance of the data augmentation step, we report the results of the evaluation metrics with and without data augmentation.

Finally, we perform cross fold validation to ensure that our reported result is complete. Because of the difference in the number of images in each dataset, we use 3-fold cross validation for the Oxford 17 and Nagy datasets and 5-fold cross validation for the Oxford 102 dataset.

4.3 Implementation details

In the segmentation FCN and classification CNN, we resize all images to $224 \times 224 \times 3$ to provide a unified and normalised set of images to pass through the networks. This also allows faster computation of convolutions and pooling [19]. All kernel sizes at the first 5 convolutional blocks in the FCN and CNN were 3×3 to provide fine-detailed features at multiple scales as suggested by [19, 34]. In the classification CNN, a 7×7 kernel size is used at blocks 6 and 7 to generate $1 \times K$ feature maps which are then mapped at block 8 to $1 \times N$ feature vector to represent the probabilistic values for the N classes. The value of N is 17 in the Oxford 17 dataset, and 102 in both Oxford 102 and Zou-Nagy datasets.

All our implementation code is written in C++ and uses the Caffe deep learning framework [53]. Training was performed on a GTX Titan X GPU with 12GB while testing was performed on a GPU and CPU to report performance measures. Training an FCN model varies between 4 to 8 hours depending on dataset size while training a classification CNN ranges between 16 to 36 hours. Testing an unseen image on the FCN takes approximately the same time to do the classification CNN. Overall, the system processes approximately 15 unseen images in one second on the GPU and 2 seconds per image on the CPU (Intel® Core i7 4GHz).

5 Results

5.1 Flower segmentation and detection

Experiments were conducted over the three datasets. Table 1 shows the mean $\pm$ standard deviation of the segmentation IoU and the detection accuracy (B_{IoU}), with and without the data augmentation step. The accuracy has improved with this step by an average of 7.5% in segmentation and 4.1% in detection.

Table 1 The overall mean (μ) and standard deviation (σ) of different flower segmentation (IoU) and detection (B_{IoU}) results on the different flower datasets, with and without data augmentation.

Dataset	Augmentation (Y/N)	Segmentation $\mu \pm \sigma$ IoU (%)	Detection $\mu \pm \sigma$ B_{IoU} (%)
Oxford 102	N	73.6 ± 15.1	81.0 ± 16.4
	Y	80.3 ± 14.7	85.9 ± 15.5
Oxford 17	N	70.1 ± 11.2	78.7 ± 12.8
	Y	79.7 ± 6.5	82.2 ± 11.5
Zou-Nagy	N	71.4 ± 16.6	77.3 ± 16.7
	Y	79.0 ± 17.2	81.3 ± 19.3

We show in Fig. 5 box overlap ($B_{overlap}$) accuracy at different thresholds on the different datasets, which shows no substantial difference between the datasets. We achieve 95% if we consider 50% threshold while a stricter threshold such as 80% achieves 81%. No image has had any $B_{IoU} < 10\%$ which means that our segmentation model will always locate part of the flower in all images we used. Therefore, because of the robustness of the FCN, all images were used in the next step, i.e., the classification CNN.

Our mean IoU segmentation accuracy is approximately 80% and is consistent on the different datasets. Other published flower segmentation methods tested their work on Oxford 17 only, such as [54–57] and presented a larger mean IoU. On the other hand, the work in [52] achieved larger mean IoU but they dropped the most complex 4 flower classes in their testing. Furthermore, we have experimented on three datasets and all images have been included in our evaluation. Figure 6 shows an example case where the detection is accurate although the segmentation accuracy is not perfect. This suggests that our detection and consequently the classification method are generally not sensitive to partial segmentation errors.

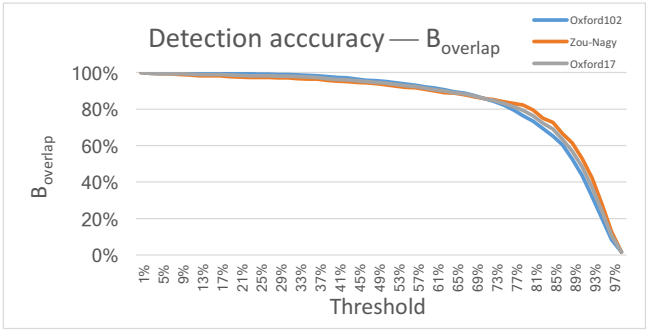

Fig. 5: Box detection accuracy at different thresholds.

5.2 Flower classification

Our CNN classifiers provide excellent results on all datasets. We achieved a classification accuracy of 99.0%, 98.5% and 97.1% on Zou-Nagy, Oxford 17 and Oxford 102 respectively. Tables 2, 3, 4 show the accuracy we achieved on the different datasets alongside recent state-of-the-art results from other groups. For each entry we report if a segmentation step is used first to localise the flower region before the classification takes place. The accuracy of the proposed method and the second best one are highlighted. We also show results with and without data augmentation to demonstrate the importance of this step.

To demonstrate the effect of the flower detection step on the accuracy of the proposed method, we report the result of flower classification using CNN with no segmentation step in the Tables. The classification CNN 'CNN only' has been trained and tested on the original images and the method shows reasonable results. However, the accuracy of the proposed FCN-CNN outperforms the 'CNN Only' method. Furthermore, having a flower segmentation and detection step is more important in classifying flower images from the Oxford datasets than the Zou-Nagy dataset; it improves the accuracy in the former dataset by 7%, and 3% in the latter.

Table 2 Flower classification accuracy on Zou-Nagy dataset.

Method	Segmentation (Y/N)	Classification accuracy (%)
FCN-CNN w/ augmentation (proposed)	Y	**99.0**
FCN-CNN w/o augmentation	Y	95.4
CNN only	N	**96.1**
CAVIAR [7]	Y	93.0
Hsu et al. [11]	Y	77.8
Saitoh et al. [54]	Y	65.5

Fig. 6: Segmentation and detection example, (a) original image, (b) manual segmentation, (c) automatic segmentation, (d) minimum bounding box of (b), (e) minimum bounding box of (c).

Table 3 Flower classification accuracy on Oxford 17 dataset.

Method	Segmentation (Y/N)	Classification accuracy (%)
FCN-CNN w/ augmentation (proposed)	Y	**98.5**
FCN-CNN w/o augmentation	Y	93.8
CNN only	N	91.4
Nilsback and Zisserman [5]	Y	88.33
FDDL-FHL [8]	Y	**97.8**
Color attention [9]	N	95.0
GHM LocSaliency [14]	N	93.5
BiCoS [15]	Y	91.1
Multi-scale fusion [17]	Y	91.39
Het. co-oc. feat. [28]	N	94.19
CEC [29]	N	93.7
GRID-FLH [30]	N	94.0
Harr-like trans. [31]	N	91.87
GRLF [32]	N	91.7
mTDP [39]	N	94.8
H-DSR [42]	N	87.1
Inception-v3 [46]	N	95.0

Table 4 Flower classification accuracy on Oxford 102 dataset.

Method	Segmentation (Y/N)	Classification accuracy (%)
FCN-CNN w/ augmentation (proposed)	Y	**97.1**
FCN-CNN w/o augmentation	Y	94.3
CNN only	N	90.6
Nilsback and Zisserman [5]	Y	72.8
CNN-RI-Deep [10]	N	94.01
TriCoS [13]	Y	85.2
PRICoLBP [16]	Y	84.2
Metric forests [25]	N	93.51
GMP [26]	N	84.6
VA [27]	N	86.31
CNN-HFL [35]	N	83.35
ONE-SVM [36]	N	86.82
CNNaug-SVM [37]	N	86.8
MsML+ [38]	N	89.45
Zheng et al. [40]	N	95.6
WTA [41]	N	83.2
Liu et al. [43]	N	84.0
CFN [44]	N	82.6
Pro-CRC [45]	N	94.8
Inception-v3 [46]	N	94.0
SCDA [47]	N	92.1
LG-CNN [48]	Y	**96.6**

Classification accuracy has improved in all datasets when using data augmentation as demonstrated in Tables 2, 3, 4. The improvement in classification accuracy is 3.6%, 4.5%, 2.8% in Zou-Nagy, Oxford 17 and Oxford 102 respectively. Oxford 102 has the least improvement because it is the dataset with the largest number of images per-class. Oxford 17 is the dataset which benefits the most although Zou-Nagy has fewer images per class. This could have happened because the variability in the Oxford 17 dataset is much larger and hence data augmentation helps make the classification of this dataset more robust.

The proposed method fails on few cases. The main reasons might be due to incorrect manual annotation, close similarity in appearance of different flower classes, and large difference in flower appearance compared to other images from the same class. Figure 7 shows two images from two classes where one image is correctly classified while the other fails. It is clear that images from the same class could vary significantly in appearance, shape and pose.

6 Conclusion

A deep learning-based method to segment, detect and classify flower images is presented in this paper. Novel ideas are demonstrated in this work which make the method robust and successful on a variety of datasets. Unlike other methods which rely on handcrafted features, the proposed method learns the most discriminative features within a deep learning framework. Segmentation and detection of the minimal flower region allows for a more accurate classification because it allows the classification CNN to focus on the region of interest while excluding non-discriminative regions.

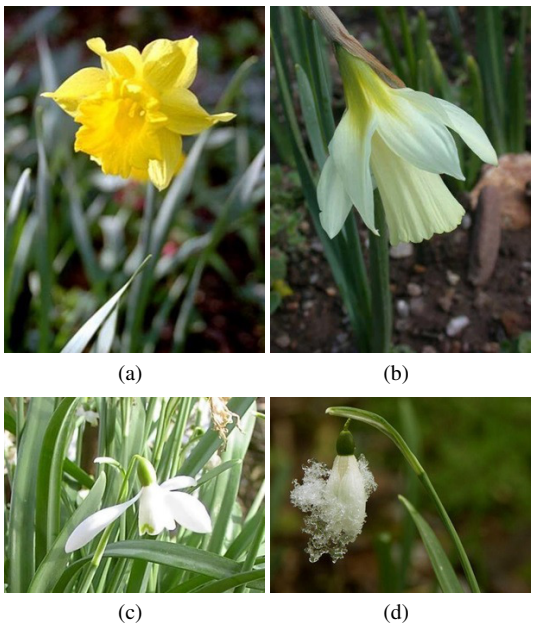

Fig. 7: Example of correctly and incorrectly classified images, (a) and (b) are from the same flower class while (c) and (d) are from another class, (a) and (c) are correctly classified while (b) and (d) are misclassified. Fig. 7(c) is cropped for better visualisation.

To our knowledge, this work demonstrates the best flower classification accuracy to-date. The main contributions which helped in achieving superior results compared to other approaches can be summarised as follows: First, the use of CNN allows a more robust classifier because it allows learning better features compared to hand-crafted features used in classical approaches. Second, localisation of the flower simplifies the classification task which means that a two-step approach is better than a one-step classification in such applications. Third, the transferred weights from the pre-trained model such as VGG-16 and consequently from the segmentation FCN to the classification CNN allows faster convergence and a more accurate solution when optimising the weights. Fourth, gradual CNN learning and avoiding local minima provide a progressive learning of the classification CNN via (1) learning low-level, mid-level and high-level layers independently then optimising all layers, and (2) an automatic restart of the optimiser by suddenly increasing the learning rate a few times during training. Finally, the proposed data augmentation step makes the CNN more robust, as demonstrated by the results. This step improves the CNN classification because it adds rotation-aware information to the CNN and it allows robust learning when the variability of flower shape, pose and appearance is huge.

We developed a good binary segmentation method for the flower region, though the main aim of this work is to propose an accurate and robust classification method. The classification accuracy approaches perfection on the three datasets. The proposed method is very accurate and only 168 out of more than 10 thousand images were misclassified from all datasets. Finally, although we show the applicability of the proposed method on the flower classification problem, our method can be applied to other applications which share similar challenges with flower classification. In addition, our proposed method might be suitable for use in applications which allow sharing, annotating and organising meaningful content in images such as Visipedia [58].

7 Acknowledgements

We would like to thank the research groups (Oxford University VGG group and Rensselaer Polytechnic Institute) who provided the datasets and the manual ground truth. Dr Mohammad Yaqub is funded by Innovate UK (Project 101684) and the UK Engineering and Physical Sciences Research Council (EP/L505316/1).

8 References

1 Kenrick, P.: 'Botany: The family tree flowers', Nature, 1999, **402**, (6760), pp. 358–359
2 Das, M., Manmatha, R., Riseman, E.: 'Indexing flower patent images using domain knowledge', IEEE Intelligent Systems and their Applications, 1999, **14**, (5), pp. 24–33
3 Larson, R. (Ed.): 'Introduction to Floriculture' (Academic Press, 2nd edn. 1992)
4 Chi, Z.: 'Data management for live plant identification', in Feng, D. *et al.* (Ed.): 'Mutimedia Information Retrieval and Management' (Springer, 2003), pp. 432–457
5 Nilsback, M., Zisserman, A.: 'Automated flower classification over a large number of classes'. Proc. Sixth Indian Conference on Computer Vision, Graphics & Image Processing, Bhubaneswar, India, December 2008, pp. 722–729
6 Nilsback, M., Zisserman, A.: 'A visual vocabulary for flower classification'. Proc. IEEE Conference on Computer Vision and Pattern Recognition, New York, NY, June 2006, **2**, pp. 1447–1454
7 Zou, J., Nagy, G.: 'Evaluation of model-based interactive flower recognition'. Proc. IEEE International Conference on Pattern Recognition, Cambridge, UK, August 2004, **2**, pp. 311–314
8 Yang, M., Zhang, L., Feng, X., Zhang, D.: 'Sparse Representation Based Fisher Discrimination Dictionary Learning for Image Classification', International Journal of Computer Vision, 2014, **109**, (3), pp. 209–232
9 Khan, F., van de Weijer, J., Vanrell, M.: 'Modulating Shape Features by Color Attention for Object Recognition', International Journal of Computer Vision, 2012, **98**, (1), pp. 49–64
10 Xie, L., Wang, J., Lin, W., Tian, Q.: 'Towards Reversal-Invariant Image Representation', International Journal of Computer Vision, 2017, **123**, (2), pp. 226–250
11 Hsu, T., Lee, C., Chen, L.: 'An interactive flower image recognition system', Multimedia Tools and Applications, 2011, **53**, (1), pp. 53–73
12 Mottos, A., Feris, R.: 'Fusing well-crafted feature descriptors for efficient fine-grained classification'. Proc. IEEE International Conference on Image Processing, Paris, France, October 2014, pp. 5197–5201
13 Chai, Y., Rahtu, E., Lempitsky, V., Van Gool, L., Zisserman, A.: 'TriCoS: A Tri-level Class-Discriminative Co-segmentation Method for Image Classification'. Proc. European Conference on Computer Vision, Florence, Italy, October 2012, **I**, pp. 794–807
14 Chen, Q., Song, Z, Hua, Y., Huang, Z., Yan, S.: 'Hierarchical Matching with Side Information for Image Classification'. Proc. IEEE Conference on Computer Vision and Pattern Recognition, Providence, Rhode Island, June 2012, pp. 3426–3433
15 Chai, Y., Lempitsky, V., Zisserman, A.: 'BiCoS: A Bi-level Co-Segmentation Method for Image Classification'. Proc. International Conference on Computer Vision, Barcelona, Spain, November 2011, pp. 2579–2586
16 Qi, X., Xiao, R., Li, C., Qiao, Y., Guo, J., Tang, X.: 'Pairwise Rotation Invariant Co-Occurrence Local Binary Pattern', IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, **36**, 11, pp. 2199–2213
17 Hu, W., Hu, R., Xie, N., Ling, H., Maybank, S.: 'Image Classification Using Multi-scale Information Fusion Based on Saliency Driven Nonlinear Diffusion Filtering', IEEE Transactions on Image Processing, 2014, **23**, (4), pp. 1513–1526
18 Krizhevsky, A., Sutskever, I., Hinton, G.: 'ImageNet Classification with Deep Convolutional Neural Networks', in Pereira, F. *et al.* (Ed.): 'Advances in Neural Information Processing Systems' (Curran Associates, Inc., 2012), pp. 1097–1105
19 Simonyan, K., Zisserman, A.: 'Very deep convolutional networks for large-scale image recognition'. Proc. International Conference on Learning Representations, San Diego, CA, May 2015, arXiv preprint arXiv:1409.1556
20 Shelhamer, E., Long, J., Darrell, T.: 'Fully Convolutional Networks for Semantic Segmentation', IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, **39**, (4), pp. 640–651
21 Girshick, R., Donahue, J., Darrell, T., Malik, J.: 'Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation'. Proc. IEEE Conference on Computer Vision and Pattern Recognition, Columbus, Ohio, June 2014, pp. 580–587
22 Girshick, R.: 'Fast R-CNN', Proc. IEEE International Conference on Computer Vision, Santiago, Chile, December 2015, pp. 1440–1448
23 Ren, S., He, K., Girshick, R., Sun, J.: 'Faster R-CNN: Towards real-time object detection with region proposal networks', IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, **39**, (6), pp. 1137–1149
24 Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: 'You Only Look Once: Unified, Real-Time Object Detection'. Proc. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada, June 2016, pp. 779–788
25 Xu, Y., Zhang, Q., Wang, L.: 'Metric forests based on Gaussian mixture model for visual image Classification', Soft Computing, 2018, **22**, (2), pp. 499–509
26 Murray, N., Perronnin, F.: 'Generalized Max Pooling'. Proc. IEEE Conference on Computer Vision and Pattern Recognition, Columbus, Ohio, June 2014, pp. 2473–2480
27 Xie, L., Wang, J., Zhang, B., Tian, Q.: 'Incorporating visual adjectives for image classification', Neurocomputing, 2016, **182**, pp. 48–55
28 Ito, S., Kubota, S.: 'Object Classification Using Heterogeneous Co-occurrence Features'. Proc. European Conference on Computer Vision, Heraklion, Crete, Greece, September 2010, **V**, pp. 701–714
29 Zhang, C., Huang, Q., Tian, Q.: 'Contextual Exemplar Classifier Based Image Representation for Classification', IEEE Transactions on Circuits and Systems for Video Technology, 2017, **27**, (8), pp. 1691–1699
30 Fernando, B., Fromont, E., Tuytelaars, T.: 'Mining Mid-level Features for Image Classification', International Journal of Computer Vision, 2014, **108**, (3), pp. 186–203
31 Zhang, C., Liu, J., Liang, C., Huang, Q., Tian, Q.: 'Image classification using Harr-like transformation of local features with coding residuals', Signal Processing, 2013, **93**, (8), pp. 2111–2118
32 Ye, G., Liu, D., Jhuo, I., Chang, S.: 'Robust Late Fusion with Rank Minimization'. Proc. IEEE Conference on Computer Vision and Pattern Recognition, Providence, Rhode Island, June 2012, pp. 3021–3028
33 He, K., Zhang, X., Ren, S., Sun, J.: 'Deep residual learning for image recognition'. Proc. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada, June 2016, pp. 770–778
34 Szegedy, C., Liu, W., Jia,Y. *et al.*: 'Going deeper with convolutions'. Proc. IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, June 2015, pp. 1–9
35 Song, G., Jin, X., Chen, G., Nie, Y.: 'Two-level hierarchical feature learning for image classification', Frontiers of Information Technology & Electronic Engineering, 2016, **17**, (9), pp. 897–906
36 Xie, L., Hong, R., Zhang, B., Tian, Q.: 'Image Classification and Retrieval are ONE'. Proc. 5th ACM on International Conference on Multimedia Retrieval, Shanghai, China, June 2015, pp. 3–10
37 Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: 'CNN Features Off-the-Shelf: An Astounding Baseline for Recognition'. Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, Ohio, June 2014, pp. 512–519
38 Qian, Q., Jin, R., Zhu, S., Lin, Y.: 'Fine-Grained Visual Categorization via Multi-stage Metric Learning'. Proc. IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, June 2015, pp. 3716–3724
39 Xie, G., Zhang, X., Shu, X., Yan, S., Liu, C.: 'Task-Driven Feature Pooling for Image Classification'. Proc. IEEE International Conference on Computer Vision, Santiago, Chile, December 2015, pp. 1179–1187
40 Zheng, L., Zhao, Y., Wang, S., Wang, J., Tian, Q.: 'Good Practice in CNN Feature Transfer', arXiv preprint arXiv:1604.00133, 2016
41 Bakhtiary, A., Lapedriza, A., Masip, D.: 'Winner takes all hashing for speeding up the training of neural networks in large class problems', Pattern Recognition Letters, 2017, **93**, pp. 38–47
42 Zhang, C., Li, R., Huang, Q., Tian, Q.: 'Hierarchical Deep Semantic Representation for Visual Categorization', Neurocomputing, 2017, **257**, pp. 88–96
43 Liu, Y., Tang, F., Zhou, D., Meng, Y., Dong, W.: 'Flower Classification via Convolutional Neural Network'. Proc. IEEE International Conference on Functional-Structural Plant Growth Modeling, Simulation, Visualization and Applications, Qingdao, China, November 2016, pp. 110–116
44 Liu, Y., Guo, Y., Lew, M.: 'On the Exploration of Convolutional Fusion Networks for Visual Recognition', in Amsaleg, L. *et al.* (Ed.): 'MultiMedia Modeling' (Springer, 2017), pp. 277–289
45 Chakraborti, T., McCane, B., Mills, S., Pal, U.: 'Collaborative Representation based Fine-grained Species Recognition'. Proc. International Conference on Image and Vision Computing New Zealand, Palmerston North, New Zealand, November 2016, pp. 1–6
46 Xia, X., Xu, C., Nan, B.: 'Inception-v3 for flower classification'. Proc. International Conference on Image, Vision and Computing (ICIVC), Chengdu, China, June 2017, pp. 783–787
47 Wei, X., Luo, J., Wu, J., Zhou, Z.: 'Selective Convolutional Descriptor Aggregation for Fine-Grained Image Retrieval', IEEE Transactions on Image Processing, 2017, **26**, (6), pp. 2868–2881
48 Xie, G., Zhang, X., Yang, W. *et al.*: 'LG-CNN: From local parts to global discrimination for fine-grained recognition', Pattern Recognition, 2017, **71**, pp. 118–131
49 Shapiro, L., Stockman, G.: 'Computer Vision' (Prentice Hall, 2001), pp. 53–54
50 Loshchilov, I., Hutter, F.: 'SGDR: Stochastic Gradient Descent with Warm Restarts'. Proc. International Conference on Learning Representations, Toulon, France, April 2017, arXiv preprint arXiv:1608.03983
51 Nilsback, M., Zisserman, A.: 'Delving into the Whorl of Flower Segmentation'. Proc. British Machine Vision Conference, Warwick, UK, September 2007, pp. 54.1–54.10
52 Nilsback, M., Zisserman, A.: 'Delving deeper into the whorl of flower segmentation', Image and Vision Computing, 2010, (28), (6), pp. 1049–1062
53 Jia, Y., Shelhamer, E., Donahue, J. *et al.*: 'Caffe: Convolutional Architecture for Fast Feature Embedding'. Proc. 22nd ACM international conference on Multimedia, Orlando, Florida, November 2014, pp. 675–678
54 Saitoh, T., Aoki, K., Kaneko, T.: 'Automatic Recognition of Blooming Flowers'. Proc. IEEE International Conference on Pattern Recognition, Cambridge, UK, August 2004, **1**, pp. 27–30
55 Aydin, D., Uğur, A.: 'Extraction of flower regions in color images using ant colony optimization', Procedia Computer Science, 2011, **3**, pp. 530–536
56 Visin, F., Romero, A., Cho, K. *et al.*: 'ReSeg: A Recurrent Neural Network-based Model for Semantic Segmentation'. Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, Nevada, June 2016, pp. 426–433
57 Liu, F., Lin, G., Qiao, R., Shen, C.: 'Structured Learning of Tree Potentials in CRF for Image Segmentation', IEEE Transactions on Neural Networks and Learning Systems, 2017, DOI: 10.1109/TNNLS.2017.2690453, pp. 1–7
58 Belongie, S., Perona, P.: 'Visipedia circa 2015', Pattern Recognition Letters, 2016, **72**, pp. 15–24