Image Classification and Segmentation of Oxford Flower Dataset using Convolutional Neural Networks (CNNs)

Rashed Alaleeli
School of Computer Science
University of Nottingham
efyra5@nottingham.ac.uk

***Abstract*—The research paper aims to explore the use of Convolutional Neural Networks (CNNs) in building image classification and segmentation networks that will be applied to the Oxford flower dataset. The dataset consists of 17 flower classes, with 80 images for each category, that was used for image classification. While image segmentation was trained only on a single flower class which is the Daffodil flower and it consists of 71 labeled images. The use of transfer learning techniques and pre-trained models were employed for image classification and achieved an accuracy of 89.22%. Alternatively, developed a CNN model to perform image segmentation and it attained an accuracy of 81.101% on unseen data.**

***Index Terms*—Classification, Darknet-53, Segmentation, Transfer Learning**

I. Introduction

Image classification is defined as the process of labeling an image, those labels are categorized according to the information collected from the images. While Image segmentation is the task of dividing an image into parts or regions which are called segments, the partitioning is based on features of the pixels. Multiple techniques can perform such tasks, but deep learning is the best for these tasks since it employs CNNs to extract and learn features from the images. The models consist of numerous layers each extracting various characteristics from the input image.

Image classification and segmentation using deep learning have been a part of Computer Vision since 1990. LeNet was initially introduced in that year and was proposed by LeCun [1], that advancement began a new era for CNNs, and numerous models were developed in the years to come. AlexNet, VGG19, ResNet-18, Inception, and Darknet-53 are some of the models that were built and trained on the LSVRC ImageNet dataset [2]. ResNet is one of the highest-performing models with the lowest error rate of 3.57 %, it was developed by K. He, X. Zhang, S. Ren, and J. Sun in 2015, the model comprises up to 152 layers consisting of convolutional layers, max pooling layer, and an average pooling layer [3]. Although, the architecture of the model is very complex making it computationally expensive and hard to train.

Transfer learning is the concept of utilizing pre-existing models that were trained on a huge dataset and retraining them on a new dataset, it can be used to perform both tasks with better performance than building a new model while saving training time. This idea was implemented in this research to execute the task of classification using the Darknet-53 network. The model was first introduced in 2018 by J. Redmon and A. Farhadi, who intended to enhance YOLO by combining YOLOv2, Darknet-19, and some of the Resnet approach, which resulted in a new network called Darknet-53 that consists of 53 different convolutional layers. The model has the same accuracy and performance as Resnet, however, it is two times faster to train and more efficient, this is because the model has fewer layers of 53 while Resnet has 150 [4].

The dataset used to train the models was provided by the University of Oxford, it consists of 17 different types of popular flowers in the UK, each with 80 images, making it a total of 1360 images, and all images have different characteristics including texture, shape, and colors [5]. All of the images will be used for the classification task, whereas only the Daffodil flower will be used for the segmentation.

Fig. 1. A selection of flower images sourced from the database provided by Oxford [6].

The research will look into the methodology of classifying and segmenting the images, which starts by organizing the given data, then pre-processing the images to split them into train and validation. Following that, models will be trained using the training data and evaluate how they perform using the validation data. For classification, image labels will be presented as a title for them, while segmentation, will be demonstrated as colored regions for each individual class.

II. Method

A. Classification

Darknet-53 will be used to perform the classification task, which takes an input image of 256x256x3. 4 random images were selected then to investigate their size, displayed them all together in a single plot with their size, as shown in Figure 2, and found that each image has a different size, while CNN requires all input images to have a similar size, and for darknet-53 images need to be resized to 256x256. Therefore, looped through each image in the datastore and resized them with the desired size then replaced each image with the one in the folder by writing them to the file specified by their name.

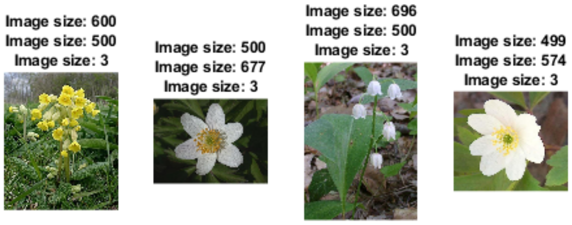

Fig. 2. Sizes of selected images from the dataset.

Data then need to be organized into subfolders, each specie should be in a separate folder. Firstly, folders need to be created using the flowers' names, but after going through the images discovered that the images were sorted randomly rather than alphabetically. For that reason, defined an array of flower names according to their order in the folder, after looping through the whole folder manually to investigate how they are ordered, then used the array to create the subfolders, those names will be used as labels for the images. Subsequently, moved each image to its corresponding folder, the first 80 images will go to the first folder, and so on, used the modulo to move to the next folder if it equals 0, then images will be moved to the next folder.

Before retraining the model, data must be split into train and validation. To do so, created an image datastore for the images with their subfolders, which will treat the folders' names as a label source. Following that, data were with a 70-30 ratio ensuring that the split is randomized, the ratio was used based on research provided by the University of Texas which says that the model performs best when a 70-80% split is used for training and 30-20% of the data is used for validation [7].

Darknet-53 was then loaded to be trained using the flowers dataset, and after analyzing the network discovered that the last three layers namely 2D convolution, softmax, and classification output layer need to be replaced since they are configured for 1000 classes, while it should have the same output number as the number of flowers species which is 17, so those layers were removed and replaced with a fully connected layer with an output size of 17 classes and a learning rate factor for weights and biases of 10. Also, added a softmax layer as the activation function that will produce the output as probabilities, and a classification layer, those 3 layers were then added and connected to the layer graph, as shown in Figure 3.

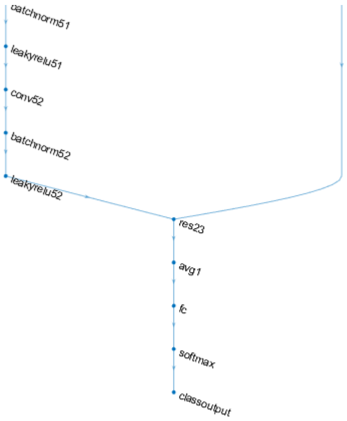

Fig. 3. Last layers of Darknet-53 after being replaced.

The network was then trained using the training data, new layers, and training options. When specifying the training options, chose a high number for the mini-batch size and a lower number for the maximum epochs to reduce the number of iterations and training time, which is calculated based on (1). Also, to speed up the learning in layers, the initial learning rate was increased to $1\mathrm{e}^{-3}$.

$$\text{\# Iterations} = \frac{\text{\# Training Samples}}{\text{Mini Batch Size}} \times \text{\# Epochs}\ , \qquad (1)$$

B. Segmentation

The data used for image segmentation consists of two folders: one for the daffodil images and another is the ground truth data. Each folder was stored in an image datastore, then 4 images were selected randomly to be checked for dimensions. It was found that all images had the same of 256x256 and do not need to be resized since it is the desired size. As shown in Figure 4, the image can be segmented into 5 different classes, such as boundaries, flowers, leaves, background, and sky. Not all of the classes were inspected because some of them were unreliable like leaves and sky, and some images do not have a region indicating the sky, and ended up inspecting the boundaries, flower, and background as ground truth values. Therefore, a pixel-label datastore was created with the labels, class names, and pixel IDs.

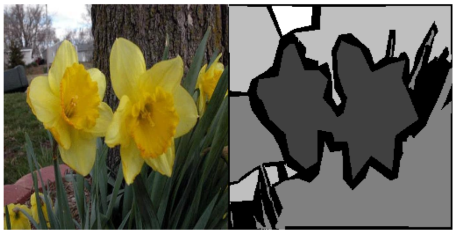

Fig. 4. A daffodil image with its segmentation ground truth labels.

Prior to model training, it is necessary to split the data into train and validation sets to evaluate the model afterward. To do so, there were no functions that support splitting the data in the way data was organized; as a result, generated random indices with 90% of the total number of images that will be used for training, and the remaining indices will be utilized for validation. Following that, divide the photos and labels based on the created indices. Then, created training data by combining the training images and labels in a pixel label image datastore, to train the semantic segmentation network.

The process of segmenting images using a CNN is divided into four major components. Starting with the input layer, which takes an input size of 256x256x3 as a parameter and applies normalization to it, the output will then be passed to the downsampling component and performs multiple operations on the input to reduce the spatial resolution of the data and extract high-level features, it contains multiple layers, the first is a 2D convolutional layer that takes a filter size of 3x3 to generalizes better and 64 filters. Following that, Relu was chosen as an activation function since it prevents the vanishing gradient on the positive side comparing it with sigmoid and tanh. To mitigate the model's sensitivity to initialization, a normalization layer was placed between the convolutional layer and Relu, and lastly, a max pooling layer that downsamples the data by taking the maximum pixel of a 2x2 patch. Downsampling is made up of 4 2D convolutional layers, 4 Relu layers, 4 normalization layers, and 2 max-pooling layers, organized in a specific pattern making a total of 14 layers. Images must then be resized to the original size, which can be accomplished through upsampling, beginning with a transposed convolutional layer to increase the resolution of images to the desired size, and a Relu as an activation function. The final layers are those for classification, starting with a softmax layer which represents the output as probabilities, then at the end is the pixel classification layer outputs the class names for each pixel value, it takes the class names to balance the process of labeling them according to their weights and avoid any imbalance since the network assume the labels are evenly distributed through the images, as in table (I), background pixels are the most frequent one and the network will consider that when assigning labels to pixels. The class weights were calculated using (2). The model overall consists of 22 layers, as shown in Figure 5.

TABLE I
PIXEL VALUES OF EACH CLASS AND THEIR WEIGHTS.

Class name	Pixel Count	Image Pixel Count	Frequency	Class Weights
Boundaries	553,790	4,194,300	0.1382	7.2345
Flowers	1,047,600	4,194,300	0.2615	3.8241
Background	2,404,900	4,063,200	0.6003	1.6659

$$\text{Class Weights} = \left(\frac{\text{Pixel count in a class}}{\text{Total number of pixels}}\right)^{-1} \qquad (2)$$

The network was then trained with a learning rate of $1\mathrm{e}^{-2}$ to speed up the process of training, 50 epochs, and a mini-batch size of 64. Nevertheless, the network was less computationally expensive since the data is smaller comparing it with the classification network.

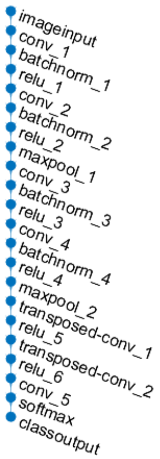

Fig. 5. Architecture of the semantic segmentation network.

III. Evaluation

A. *Classification*

The classification network was hard to train and computationally expensive, due to the high number of training data. However, it achieved a high accuracy of 89.22% when testing it on the validation data. As shown in Figure 6, these are random validation images selected to be classified, and after investigating their labels, all of them seem to be correctly labeled.

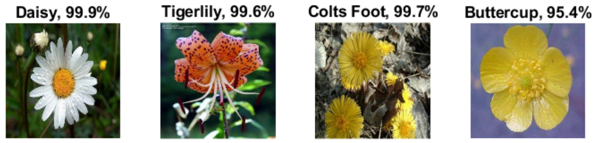

Fig. 6. Classification result of the retrained network.

When classifying all the flowers, the model shows varied levels of accuracy for each flower, where some flowers are more accurately classified comparing it to others. As shown in Figure 7, the accuracy of labeling Sunflowers, and Colts Foot correctly is equal to 100%, this is maybe because these flowers have a unique shape differentiating them from the others, making the model easily classifies them. Nevertheless, the model has a higher error rate for classifying some flowers, such as Crocus and Tulip, this could be because of the similarities between these classes.

B. *Segmentation*

Training the segmentation network was faster comparing it with the classification network due less images being trained

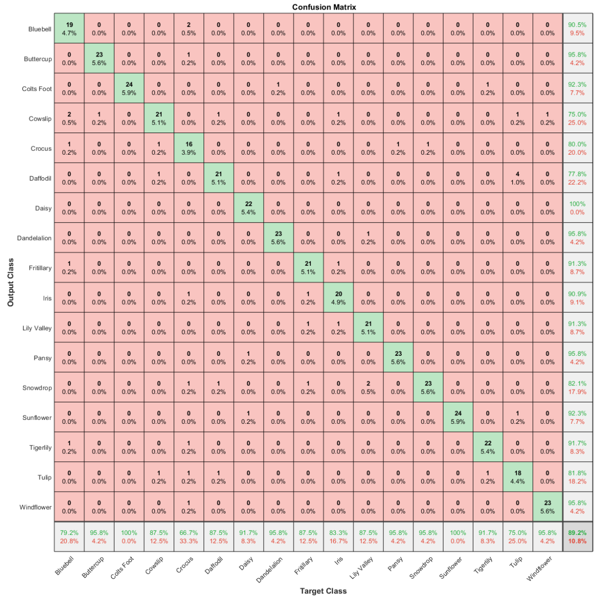

Fig. 7. Confusion matrix of the classification network.

making the number of iterations lower. The network performed an accuracy of 88.09% with a loss of 0.5148 on the training data. However, to evaluate the model it must be tested on unseen data, to perform that, segmented the validation images and predicted their class labels, then compared them with the original labels, and it achieved a mean accuracy of 81.101%. Subsequently, a confusion matrix was utilized to check which labels were predicted correctly, as shown in Figure 8, the network seems to segment the background correctly with an accuracy of 95.1%, while segmenting the flowers has a lower accuracy of 91.7% due to segmenting some of the flowers as a boundary. However, the model has low performance in predicting the boundaries with an accuracy of 56.5%, while the other 43.5% the model segmented the boundary as flowers and background.

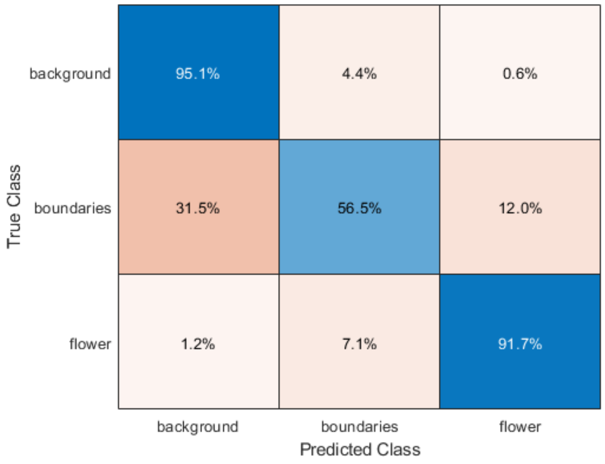

Fig. 8. Confusion matrix of semantic segmentation network.

A sample result of the predicted segmentation is shown in Figure 9, the model seems to not segment all parts of the background. In contrast, the model seems to segment the flowers perfectly as it segmented the blurred flower correctly while the validation image does not, this is maybe due to the model that generated the ground truth values being imbalanced and did not consider the class weights when training the models, this can be a source of error which made the accuracy on validation images lower than expected.

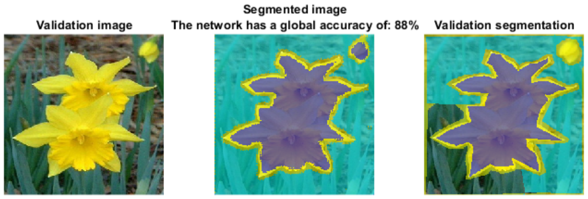

Fig. 9. Comparing the segmentation results on a validation image.

IV. Conclusion

In this research, explored the usage of transfer learning by retraining Darknet-53 on the flowers dataset and the results indicate that the network can highly classify the flowers according to the features extracted. On the other hand, the segmentation network's performance was lower since it was built from scratch, which gives preference to the use of transfer learning. However, some possible steps that could be taken for the networks to achieve a high degree of accuracy which include, data augmentation, and increasing the complexity of the models as deeper networks may extract more features and increase the performance of models. Additionally, increasing the training data can have a positive impact on the performance of the model, the usage of generative adversarial networks (GANs) can be implemented to generate more flower images, allowing the model to be trained on a range of a variety of images, but those networks are tough to train.

References

[1] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, “Deep Learning for Computer Vision: A brief review,” Computational Intelligence and Neuroscience, vol. 2018, pp. 1–13, Feb. 2018.

[2] I. Naseer, S. Akram, T. Masood, A. Jaffar, M. A. Khan, and A. Mosavi, “Performance analysis of state-of-the-art CNN architectures for luna16,” Sensors, vol. 22, no. 12, p. 4426, 2022.

[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[4] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” arXiv.org, 08-Apr-2018. [Online]. Available: https://arxiv.org/abs/1804.02767. [Accessed: 05-May-2023].

[5] M.-E. Nilsback and A. Zisserman, “A visual vocabulary for Flower Classification,” 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06), Oct. 2006.

[6] X. Piao, Y. Hu, Y. Sun, J. Gao, and B. Yin, “Block-Diagonal Sparse Representation by Learning a Linear Combination Dictionary for Recognition. ,” p. 11, Jan. 2016.

[7] A. Gholamy, V. Kreinovich, and O. Kosheleva, ”Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation” (2018). Departmental Technical Reports (CS). 1209.