Research on Image Classification and Segmentation using Matlab

HoHim Lee[1, *]

[1] *School of Physics and Astronomy, University of Nottingham, Nottingham, NG7 2RD, UK*

(Dated: May 16, 2023)

In this project, we have performed image classification and segmentation on the 17 flowers dataset [1]. The dataset includes 17 classes of flowers for classification and a set of flower images with the ground-truth segmentation map for segmentation.

I. Introduction

In recent decades, Computer Vision has been one of the most important research areas for computer and machine learning scientists, it gives machines or computers the ability to recognise and understand visual data, which introduces endless possibilities of use cases and applications, such as self-driving car, medical image analysis and facility automation. There are two main goals for Computer Vision, the first is to create computational models of the human visual system from a biological science perspective, and the other is to create autonomous systems that are capable of carrying out certain tasks that are within the limit of the human visual system or even outperform it. [2]

The idea of Computer Vision was first introduced by Larry Roberts in his PhD thesis titled Machine Perception of three-dimensional Solids [3]. In his research, he discussed the possibilities of extracting information from 2D perspective views, and this lay the foundation of modern days Computer Vision research.

Currently, there are two main tasks for Computer Vision, which are Semantic Segmentation and Image Classification. The objective of a segmentation task is to categorize groups of pixels in an image into a class or object by generating a segmentation map, fig. 1 shows an example of a Segmentation map overlay on an image consisting of a flower and attempts to separate the flower from the background. The objective of Image Classification is to classify the entire image by assigning it to a specific label.

The goal of this project is to perform Image Classification and Semantic Segmentation tasks using different machine-learning techniques, the dataset used in this project will be the 17 Category Flower Dataset from the University of Oxford [1]. This dataset contains 17 categories of flowers and each with 80 images, the ground-truth maps for performing segmentation maps were also included in this dataset, but only a subset of 1-class of the segmentation images and the corresponding segmentation ground-truth maps will be used in this project. Both of these tasks will be implemented using Matlab, an industry-spec programming language which provides many useful toolboxes to handle images and implement machine learning models.

* ppxhl1@nottingham.ac.uk

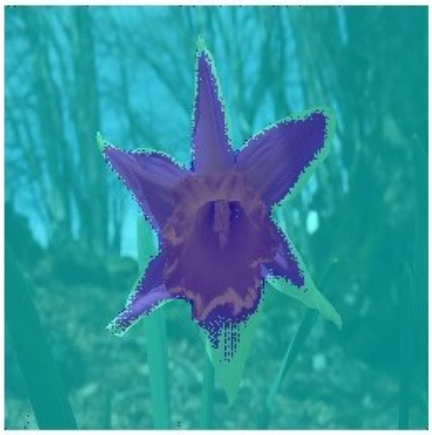

FIG. 1. Shows an example of a Image Segmentation tasks.

II. Method

A. Image Classification

Two different approaches were used to perform the Image Classification task, the first being building a custom model from the ground up and the second one is to use the AlexNet pre-trained model [4]. This is to compare the performance difference between using a pre-trined network and a custom network.

The process of loading and preparing the data is the same for both the pre-trained and the custom network. All the images were stored in the 17flowers folder, a for loop is used to go through each file in the folder and add the full file path to a cell array. After that, in order to correctly label all the images, a matrix variable with 17 rows and 80 columns is created, where each row corresponds to a different type of flower and each column corresponds to a different image of that flower. The "repmat" function is used to replicate the 1:17 vector 80 times to create the 17x80 matrix. Then, the matrix is reshaped into a column vector with 1360 rows using the colon operator. Finally, an "imageDataStore" variable is created to store all the images and their corresponding labels.

The original size of the images varies from one image to another, therefore it is needed to resize all the images to the same size before passing them into the networks. A custom function [5] is used to resize all the images to the

correct size, for the custom network, all the images were resized to 256-by-256. For the AlexNet [4] pre-trained network, all the images were required to be resized to 227-by-277 to match the AlexNet model input layer size of 227-by-277.

Overfitting means a model becomes too complex and starts to memorize the training data instead of learning the underlying patterns, this can be detected by comparing the validation accuracy and the training accuracy, if the training accuracy is substantially higher than the validation accuracy, the model is overfitting. Small datasets often have limited variations and are more susceptible to overfitting, in this case, there are only 80 images for each of the classes, which is generally not enough to train a complex CNN network with a lot of layers, therefore, the custom model is kept as simple as possible.

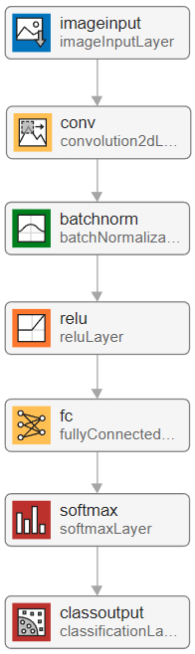

FIG. 2. Shows the CNN architecture for the custom model.

The CNN architecture for the custom model is shown in fig. 2, it first starts with an input layer which defines the input size of the image to be [256 256 3]. The next layer will be a 2-D convolutional layer which will apply 20 convolution filters of size 5x5 to the input image, a 2-D convolutional layer is used to extract features from 2D image data through a process called convolution. The output of the 2-D convolutional layer will then pass into a batch normalization layer which will normalize the activations of the previous convolutional layer over a mini-batch of data. It helps in improving the training stability and convergence of the network. Next, a ReLU activation function is applied and then passes into a fully connected layer, this layer connects all the activations of the previous layer to every neuron in the output layer. The number of neurons in this layer is equal to the number of classes, which is 17 in this case. A softmax layer is then applied which converts the output into probabilities for each class. The final layer will be the classification layer which computes the cross-entropy loss between the predicted class probabilities and the true class labels.

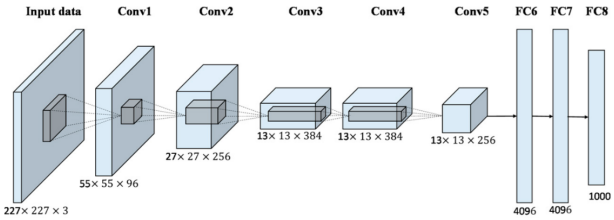

FIG. 3. Shows the CNN architecture of AlexNet model.

For the pre-trained CNN architecture, it uses the pre-trained layers from the AlexNet model shown in fig. 3, except the last layer was replaced to match the number of classes, which is 17. AlexNet is a convolutional neural network that is 8 layers deep and trained on more than a million images.

In order to compare the result of the two CNN networks, the training options are kept the same, both of them are trained on 10 epochs with a learning rate of 1e-4 and the batch size set to 16. Both models will be tested on unseen data, which will be the test set and the classification accuracy will be used as the performance matrix to compare the results.

B. Image Segmentation

One custom model was developed for the segmentation task to separate two classes, flower and background. The dataset contains a total of 71 images with the corresponding ground truth segmentation maps, 15 images were manually picked out from the dataset to be used as unseen data to elevate the model. An “imageDatastore” is used to store the training images and a “pixelLabelDatastore” is used to store the ground truth labels, they will be combined before passing into the model to train.

The architecture of the network can be split into three main parts: downsampling, upsampling, and final output layers. Two 2-D convolutional layers and one max pooling layer are in the downsampling part, they are used to reduce the dimensions of the input images while increasing the number of feature maps, this helps the model to capture more high-level features and reduce computational complexity. The upsampling part uses two transposed convolutional layers to upsample the feature maps and gradually recover the spatial dimensions of the image. The final output layers consist of a 1x1 convolutional layer, a softmax layer, and a pixel classification layer. The 1x1 convolutional layer reduces the number of feature maps to the number of classes, which will be 2, flower and background.

The model will be tested on the test set which is not used in the training process to see if the model can generalize to unseen data, the performance matrix to evaluate the model will be the MeanIoU score.

III. Evaluation

A. Classification Results

After training both the custom and pre-trained network on the training set, we can see that the custom model is overfitting in the early stage of training, fig. 4 shows the training process of the custom model, the training accuracy has become much higher than the validation accuracy after only 2 epoch, while the validation accuracy for the pre-trained model is relatively close to the training accuracy which is shown in fig. 5. It is clear that the pre-trained model is performing much better than the custom model in the training process, this can be due to the pre-trained model has been trained on a very large dataset compared to the custom model has only 80 images for each class to work with, therefore, the pre-trained model can do a much better job at extracting features from the images.

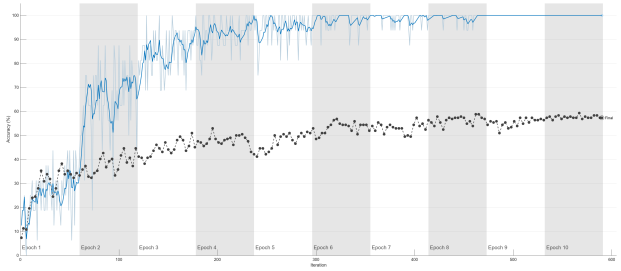

FIG. 4. Shows the training process of the custom CNN

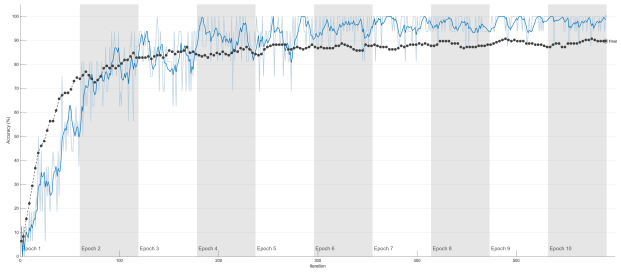

FIG. 5. Shows the training process of the pre-trained CNN.

Both models are tested on the test set to see how well they can generalize to unseen data. Across 17 classes, the custom model managed 57.4% classification accuracy while the pre-trained model managed 92.2% classification accuracy. The confusion matrix for both models are shown in fig. 6 and fig. 7. What we can see from these results is that the pre-trained model can achieve much higher accuracy not only during training but also on unseen data. Although the performance of the custom model is worse than the pre-trained model, considering there are 17 classes to classify on, over 50% accuracy for a simple model is still excellent given the limited amount of data available.

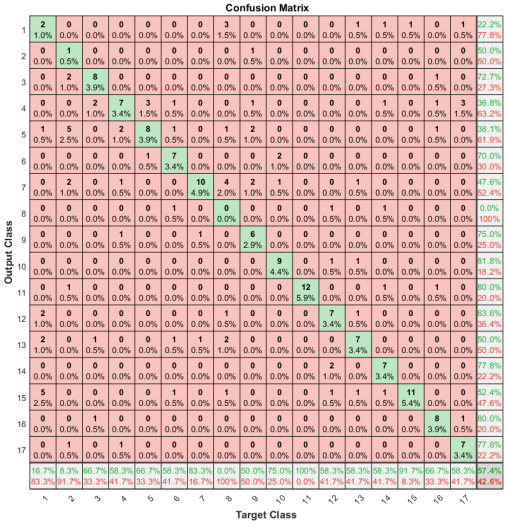

FIG. 6. Shows the confusion matrix for the custom CNN on the test set.

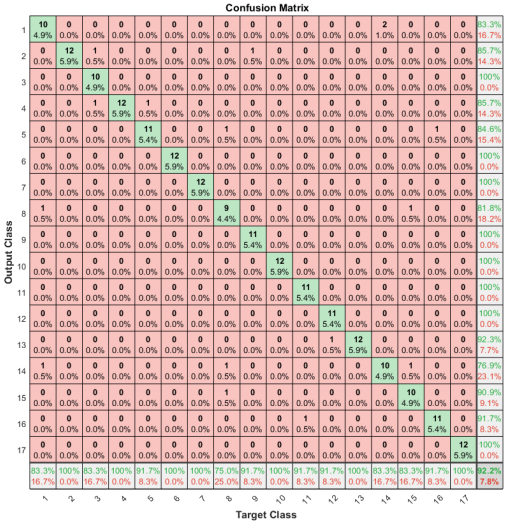

FIG. 7. Shows the confusion matrix for the pre-trained CNN on the test set.

B. Segmentation Results

The model for the image segmentation task has been trained on 100 epochs while using the SGD optimizer with the mini-batch size set to 8. It achieved over 90% accuracy on the training set. The model is tested on the test set containing 15 images to see if the model can generalize to new unseen data.

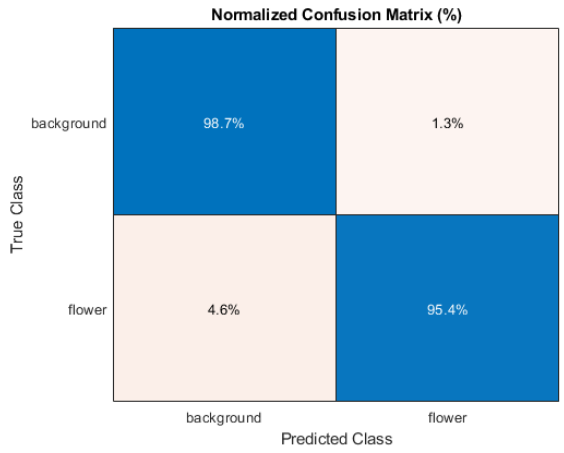

FIG. 8. Shows the confusion matrix for the segmentation model on the test set.

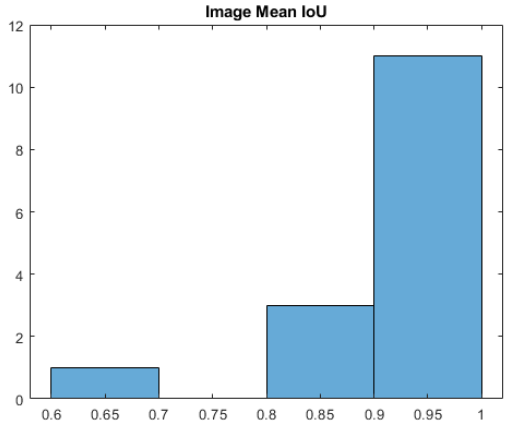

FIG. 9. Shows the mean IoU distribution for the segmentation model on the test set.

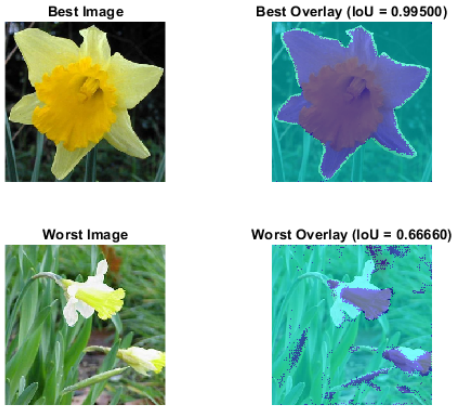

FIG. 10. Shows the image with the best and worst IoU scores.

A confusion matrix for the test data is shown in fig. 8, we can see that the model managed to achieve over 95% accuracy on the two classes, which means the model has the ability to partition an unseen image into flower and background. The distribution for the Intersection-Over-Union (IoU) is also shown in fig. 9, with only one out of 15 images the model is struggling to partition. We can see the example of the best-performing image and the worst-performing image in fig. 10. Although the model is performing well in this test set, due to the limited test data, it is unsure that the model has the ability to generalize to a large number of unseen images, therefore, further testing with more data will be required.

IV. Conclusions

In this project, we have performed image classification and segmentation on the 17 flowers dataset [1]. The dataset includes 17 classes of flowers for classification and a set of flower images with the ground-truth segmentation map for segmentation. One custom model and one model using the AlexNet pre-trained network were developed for classification, after analyzing the results, it is clear that the model with AlexNet performs much better than the custom model. One model is developed for segmentation, the result shows that the model is performing well on a limited amount of test data.

[1] M.-E. Nilsback and A. Zisserman, 17 category flower dataset, .
[2] T. Huang, Computer vision: Evolution and promise, (1996).
[3] L. G. Roberts, *Machine perception of three-dimensional solids*, Ph.D. thesis, Massachusetts Institute of Technology (1963).
[4] MathWorks, Alexnet convolutional neural network.
[5] V. L. Chagi, How can i resize image stored in imagedatastore.