RMIT
UNIVERSITY

Group Machine Learning Project Report

COSC2753 - Machine Learning

Assignment 2

Team Members:

Le Minh Quan - s3877969

Ly Tin Kiet - s3755692

Nguyen Nguyen Phong - s3904419

Table of contents

Introduction

Recently, due to the growth of technology and computing power, machine learning has become a viable solution in solving real world problems. In this project, we will tackle the problem of classifying flower types and recommending similar flower images based on the provided image. For the first task, we used a supervised learning model for the classification problem. For the second task, we used an unsupervised learning model for the recommendation system.

Literature Review

Tacking a problem similar to task 1, Yuanyuan Liu et al. proposed a convolutional neural network (CNN) framework to tackle the flower classification problem, which is similar to our first task of this report. Their dataset consisted of 63442 flower images from the Internet, divided into 79 species, and manually filtered 10667 unrelated images. A preview of their dataset is in appendix F. Their model framework is a CNN consisted of an input layer of 100x100x3, 5 convolutional layers, 4 max-pooling layers, and 3 fully connected layers, visualized in appendix E. The model achieved 76.54% accuracy, which shows that their model can classify flower species very well.

Z. S. Younus et al. suggested a content-based picture retrieval system based on Particle Swarm Optimization (PSO) and the K-means clustering technique for task 2 [2]. The PSO algorithm optimizes the initial centroids of the K-means algorithm, which organizes images based on feature similarity. The authors use Gray-Level Co-occurrence Matrix (GLCM) and Color Moment (CM) approaches to extract color and texture data from photos. The authors also conduct a sensitivity analysis to assess the impact of various parameters on system performance. The results show that the suggested system is resistant to parameter changes and can achieve high retrieval accuracy with a variety of parameter settings.

Approach

Task 1

1. Exploratory Data Analysis (EDA)

The dataset provided for this problem includes a list of flower images of different types. There are 8 types of flowers in total: Babi, Calimero, Chrysanthemum, Hydrangeas, Lisianthus, Pingpong, Rosy, and Tana. Flower image size varies from around 160 to 320 pixels and they are in RGB format.

Since there are no CSV files for features, we only have the flower images as input for our machine-learning model. The first step to cleaning the data is to check for image duplication throughout all the folders of classes. Duplicate images can introduce bias and redundancy, compromising the integrity of the dataset. By identifying and removing duplicates, the dataset becomes more representative and reliable for later analysis. We removed the duplicate images

from the data frame. After that, we manually checked the image quality and found that the images have a lot of background noises, such as humans, text, different types of flowers in the same image, etc. However, since the dataset is not large, we used all the provided images for training, validating, and testing. We then checked the class balance of the flower types and found that the dataset is imbalanced, where some flower types have more than 700 samples, whereas some only have less than 400 samples. We will tackle the problem of the imbalanced dataset in the data processing part.

2. Data Processing

There are two approaches for balancing data: oversampling and undersampling. For this problem, we perform oversampling on the minority classes. In order to oversample image data without causing too much overfitting, we used data augmentation on the images. First, we will split the images into training sets (80%), validation sets (10%), and test sets (10%). Training image files will be sampled to be equal in frequency in the data frame so that data augmentation can be performed equally in all classes. Then, data augmentation will be applied to the training set to generate a new training set for the model. The data augmentation includes a rotation range of 90 degrees, a width shift range of 25%, a height shift range of 25%, a zoom range of 25%, a shear range of 10%, and a horizontal flip. Visualization of data augmentation is provided in Appendix D. The image's color value will also be divided by 255 for the model to perform math calculations. After data augmentation, we get our training set of images in the size of 160x160, and these images will be sent by a batch of size 32 to train the model. The average image size from our dataset is 224x224, however, due to personal GPU VRAM limit (8 GB), our model cannot use input size as the average image size, therefore we used 160x160, which is the minimum image width and height.

3. Model Selection

The main type of model to apply in this problem is CNN. CNN is a deep learning method that performs well in computer vision areas, including image classification and recognition. A CNN normally has 5 layers: an input layer, a convolutional layer, a pooling layer, a fully connected layer, and an output layer. For this task, we chose 2 well-known CNN architectures, trained them on our flower dataset and picked the best model. The models we selected are AlexNet and ResNet50.

3.1. AlexNet

AlexNet is the CNN architecture that won the ImageNet large-scale visual recognition challenge in 2012. The architecture consists of eight layers: five convolutional layers and three fully-connected layers. The model used Rectified Linear Units (ReLU) activation in all the layers except for the output layer, which uses softmax activation. In this flower classification task, we modified the model to increase the performance against the flower dataset, since the original architecture is used to train a large ImageNet dataset. The original AlexNet has an input size of 227 x 227 x 3, 5 convolutional layers, 3 max pool layers, 3 fully connected layers and dropout layers of rate 0.5 to prevent overfitting. The output of the original model has 1000 neurons for 1000 classes [3]. The modified AlexNet has an input size of 160x160x3, the same number of

convolutional layers but a different filter size, 5 max pool layers, 4 fully connected layers and dropout layers of rate 0.4. The output of the modified model has 8 neurons for 8 classes. In addition, L2 regularisation is applied in the final layer to reduce overfitting. This model used Stochastic Gradient Descent (SGD) optimizer and categorical cross-entropy loss function.

3.2. ResNet50

Residual Network (ResNet) is the CNN architecture that won the ImageNet challenge in 2015. ResNet is an extremely deep neural network with 152 layers, compared to other CNN architectures such as AlexNet with 8 layers and GoogleNet with 22 layers. ResNet is the first CNN architecture that introduced the concept of skip connection. Skip connection allows the neural network to be very deep while reducing the vanishing gradient problem. This concept is implemented in either the identity block or the convolutional block [4]. In this task, we will implement a ResNet-50 which is a smaller version of ResNet 152. The implementation of ResNet-50 follows this architecture: CONV2D -> BATCHNORM -> CONV2D -> BATCHNORM -> MAXPOOL -> CONVBLOCK -> IDBLOCK*3 -> CONVBLOCK -> IDBLOCK*1 -> CONVBLOCK -> IDBLOCK*2 -> AVGPOOL -> TOPLAYER. The input size for this model is 160x160x3, and the output layer has 8 neurons for 8 classes. This model used the Adam optimizer and categorical cross-entropy loss function.

4. Ultimate Judgement and Evaluation

Since this task is a multi-class classification problem with imbalanced data, we will evaluate our model performance mainly based on the f1-score metric. We will also check out the model's accuracy and loss. The best weight for each model will be obtained using Early Stopping with patience 20, and the monitor value is the f1 validation score.

From the basic metrics comparison of the models from Appendix C and the notebook output, we can see that the modified AlexNet model outperforms the ResNet50 model in many aspects:

- Higher training f1 score (0.87 > 0.85).
- Higher validation and test f1 score (validation f1 0.76 > 0.72 and test f1 0.75 > 0.67).
- Lower validation and test loss (validation loss 0.90 < 1.11 and test loss 0.87 < 1.23).

Moreover, there are other things to consider, such as training speed, model complexity, model generalization, etc. The modified AlexNet model has a faster training speed than the ResNet50 model (40s/epoch > 66s/epoch) since it is less complex, using only 8 layers in total. Although both models have small signs of overfitting on the training set according to Appendices A and B, the modified AlexNet generalized better on unseen test data, as the difference in validation and f1 score of the modified AlexNet model is lower than the difference in validation and a test score of the ResNet50 model. The only drawback of the modified AlexNet model is it requires a lot of GPU memory to train, and the exported file size is larger. Therefore, we selected the modified AlexNet model as the final model for the flower classification task.

After selecting our best model, we will also use it to compare with the model proposed in [1] to benchmark our model. Since we use the f1 score as our metrics and [1] used accuracy, we can

assume that our model performance (0.75 f1) is at least similar to theirs (0.76 accuracy). Moreover, the flower images from our dataset have a lot more noise compared to their collected dataset, compared in Appendix E. We can conclude that our modified AlexNet model is more successful for this flower classification task.

Task 2

1. Data Processing

The two tasks use the same dataset, therefore the EDA process is the same, and repeating it here would be redundant. However, some different methods are being used for data processing in Task 2 compared to Task 1. Even though the dataset is extremely imbalanced, oversampling the images is not necessary for task 2, because it would create multiple duplicates and affect the predictions, where several duplications would appear in the 10 images recommended. Image resize would still be performed, with the width and height being 160 x 160 for consistency.

A part of the CNN model in task 1 is used for extracting features, where the model output layer is removed, leaving the last layer which is the fully connected layer with an output of shape (1, 1000) as features. After extracting features, we need to apply a dimension reduction method to reduce the number of variables so that the model is more efficient. The two common dimensional reduction methods are PCA and t-SNE. While PCA is fantastic, it does have some downsides. One of the most significant disadvantages of PCA is that it does not retain non-linear variance, while t-SNE is a non-linear dimensionality reduction approach that works well for embedding high-dimensional data into lower-dimensional data for data visualization [5]. Appendices G and H show the visualization of clusters using t-SNE and PCA.

Next, our team decided that the main type of model to apply to this problem would be a clustering model, with K-Means and DBSCAN being the selected algorithms.

2. Model Selection

2.1 K-Means algorithm

K-means clustering is a basic and widely used unsupervised machine learning technique. Cluster analysis is a data mining and machine learning approach for grouping comparable items into clusters [6]. The goal of K-means clustering is to partition a set of objects into K clusters in such a way that the total of the squared distances between the items and their assigned cluster mean is minimized. The quality of the clustering results can be influenced by the initial centroid placement. Random initialization can lead to different results in different runs. To mitigate this, it's common to perform multiple runs of K-means with different initializations and select the clustering solution with the lowest overall sum of squared distances or highest silhouette score. The best K value can be decided through the elbow method [6].

The elbow method works by first running the K-means algorithm for a range of K values (e.g., from 1 to 10) [7]. For each K value, calculate the sum of squared distances between each data point and its assigned cluster centroid. This measure is also known as the within-cluster sum of

squares (WCSS). The line graph where the x-axis represents the K values and the y-axis represents the corresponding WCSS values. Look for the "elbow" point on the graph, which the graph moves almost parallel to the X-axis from this point. The elbow point indicates a good balance between the number of clusters and the compactness of the clusters. The K value at the elbow point is considered the optimal number of clusters for the given dataset [7].

2.2 DBSCAN algorithm

DBSCAN is a clustering algorithm that defines clusters as continuous regions of high density and works well if all the clusters are dense enough and well separated by low-density regions. In the case of DBSCAN, instead of guessing the number of clusters, it will define two hyperparameters: epsilon and minPoints to arrive at clusters. Epsilon is the distance measure that will be used to locate the points/to check the density in the neighborhood of any point. minPoints is the minimum number of points (a threshold) clustered together for a region to be considered dense [8].

After selecting suitable values for epsilon and minPoints, initialize the algorithm by choosing an unvisited data point. Create a new cluster and allocate the point and its neighbors to it if the count exceeds MinPts. Mark the location as visited. Iterate over the remaining unvisited locations, repeating the neighbor and cluster-building processes for each. Label a point as noise or an outlier if it has fewer than MinPts neighbors. Continue until all of the points have been visited. The resulting clusters are formed based on point density and connectivity.

3. Ultimate Judgment and Evaluation

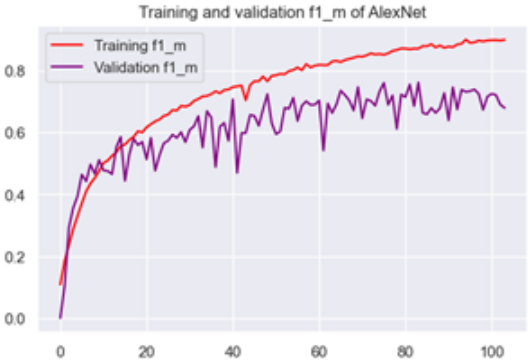
It can be said that the silhouette coefficient is one of the most popular metrics being used to compare cluster algorithms. The silhouette coefficient runs from 1 to 1, with a greater silhouette coefficient indicating a model with more cohesive clusters. Coefficients near +1 indicate that the sample is remote from the nearby clusters. A value of 0 indicates that the sample is on or near the decision border between two nearby clusters, and negative values suggest that the samples may have been allocated to the incorrect cluster [9].

Regarding the comparison of the two models in our project, the silhouette score of the K-Means algorithm is 0.45, while DBSCAN is -0.5. Since the silhouette coefficient of the DBSCAN model is negative, it is clear that the K-Means algorithm is more suitable to tackle this problem. For the recommendation result, after applying the K-Means algorithm, it seems like the recommended images are decided based on flower types, as shown in Appendix I.

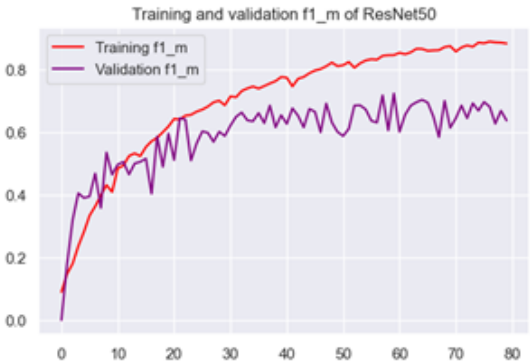
Even though the results for task 2 look promising, there are several different methods and improvements that we could have done. First, our approach for this task is almost completely different to what is suggested in the literature review paper, apart from the K-Means algorithm being used. We applied a CNN model to extract features before applying K-Means, while Z. S. Younus et al. used PSO algorithm to optimize the initial centroids for better clustering. Instead, the authors use GLCM and CM approaches to extract color and texture data. Following the path proposed in the reviewed paper could maybe produce a better image recommendations to our approach.

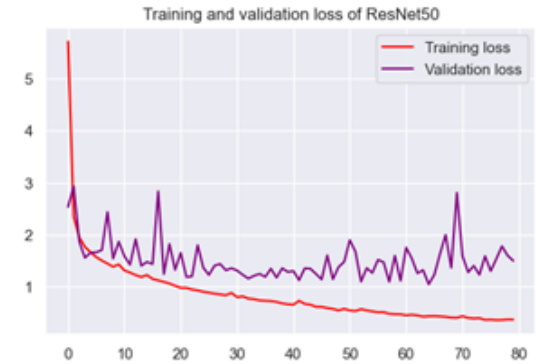
Appendices

Appendix A: Training and validation f1 score, loss of AlexNet



Appendix B: Training and validation f1 score, loss of ResNet50

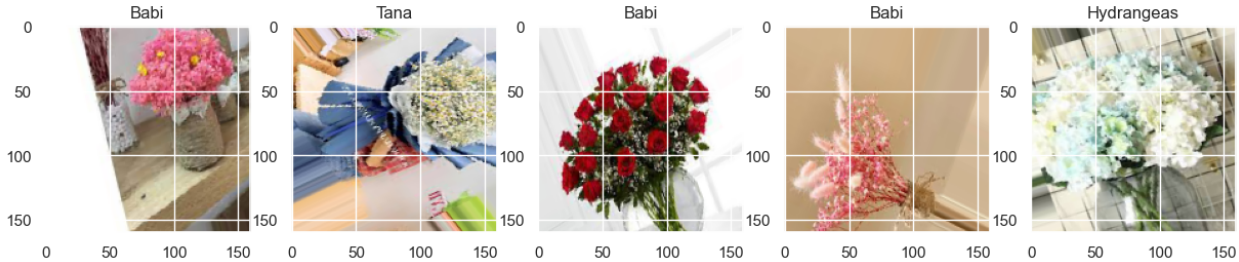




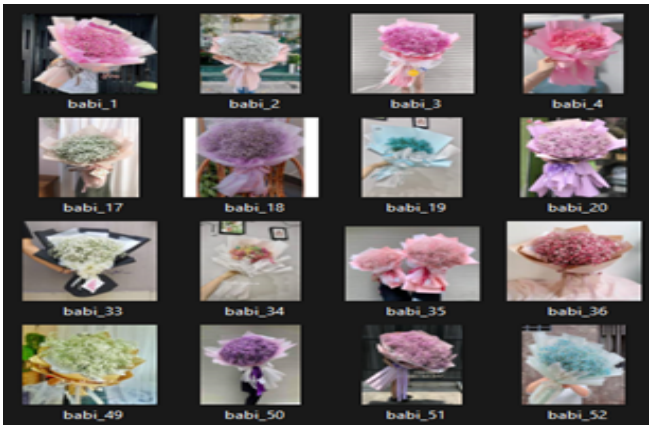
Appendix C: Comparison of AlexNet and ResNet50

	AlexNet	ResNet50
Loss	0.873385	1.231482
Accuracy	0.732719	0.672811
F1	0.747468	0.673355

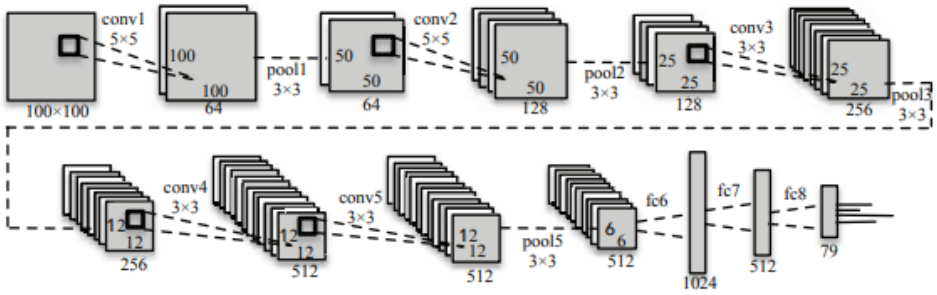
Appendix D: Data augmentation



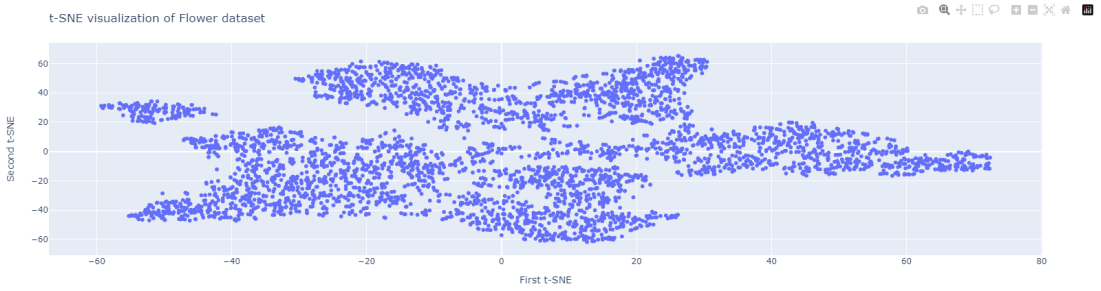
Appendix E: Dataset of [1] (left) and our dataset (right)



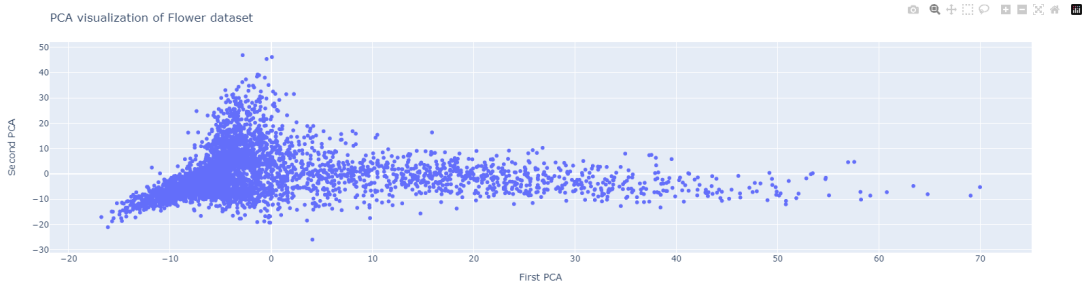
Appendix F: CNN architecture proposed by [1].



Appendix G: t-SNE visualization of the flower features



Appendix H: PCA visualization of flower features



Appendix I: Results of recommended images

Reference

[1] Y. Liu, F. Tang, D. Zhou, Y. Meng, and W. Dong, 'Flower classification via convolutional neural network', in *2016 IEEE International Conference on Functional-Structural Plant Growth Modeling, Simulation, Visualization and Applications (FSPMA)*, 2016, pp. 110–116.

[2] Z. S. Younus et al., "Content-based image retrieval using PSO and k-means clustering algorithm," Arabian journal of geosciences, vol. 8, no. 8, pp. 6211–6224, 2015, doi: 10.1007/s12517-014-1584-7.

[3] S. Saxena, "Introduction to the architecture of Alexnet," Analytics Vidhya, https://www.analyticsvidhya.com/blog/2021/03/introduction-to-the-architecture-of-alexnet (accessed May 13, 2023).

[4] P. Dwivedi, "Understanding and coding a ResNet in Keras," Medium, https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33 (accessed May 13, 2023).

[5] A. GUPTA, "PCA vs T-Sne," Kaggle, https://www.kaggle.com/code/agsam23/pca-vs-t-sne (accessed May 19, 2023).

[6] DeepAI, "K-means," DeepAI, https://deepai.org/machine-learning-glossary-and-terms/k-means (accessed May 19, 2023).

[7] A. Gupta, "Elbow method for optimal value of K in kmeans," GeeksforGeeks, https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/ (accessed May 19, 2023).

[8] Ginni, "What is DBSCAN," Tutorials Point, https://www.tutorialspoint.com/what-is-dbscan (accessed May 19, 2023).

[9] H. Belyadi and A. Haghighat, "Silhouette coefficient," Silhouette Coefficient - an overview | ScienceDirect Topics, https://www.sciencedirect.com/topics/computer-science/silhouette-coefficient (accessed May 19, 2023).