Regularization, Data augmentation and transfer learning

Ioh Nishijima
ioh.nishijima@etu.univ-st-etienne.fr
Jean Monnet University, France
Yong Thorvue
yong.thorvue@etu.univ-st-etienne.fr
Tatsuto Yamauchi
tatsuto.yamauchi@etu.univ-st-etienne.fr

1 Objective

Convolutional Neural Networks (CNNs) have become the state-of-the-art approach for various image recognition and computer vision tasks, including object detection, segmentation, and classification. However, overfitting, insufficient data and low generalization ability are the common problems that hinder the performance of CNNs.

Regularization techniques, such as L1 and L2 regularization, are commonly used to prevent overfitting. Data augmentation, such as flipping, rotating, and scaling images, can increase the size and diversity of the training data and improve the robustness of the model to variations in the input. Transfer learning, on the other hand, involves leveraging pre-trained CNN models to improve the performance of new models on related tasks.

The aim of this study is to implement a CNN to achieve optimal outcomes in a classification tasks.The intention is to assess the various methodologies explored during the lecture and evaluate their effectiveness in relation to the designated dataset.

2 Dataset

The dataset consists of ten distinct categories of flowers, with a total of 800 color images. Specifically, the training set comprises 600 images (60 per category), while the validation set consists of 200 images (20 per category). Notably, the train and validation split are pre-determined. Given that the images exhibit variable sizes, it is required to resize them uniformly to a resolution of 128x128 for the purposes of this exercise.

3 CNN model

The CNN model consists of five layers denoted by C1, C2, C3, F1, F2. The first three layers are convolutional layers. The last two layers are fully collected layers. The last layer F2 is the output of the model which has ten dimensions as following Figure 1.

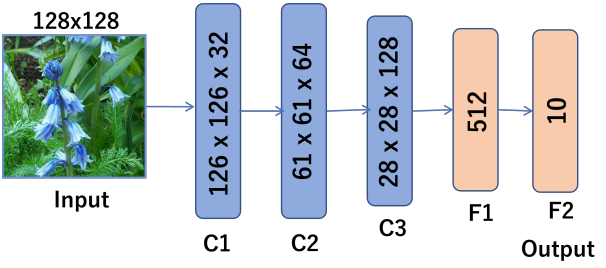

Figure 1: The architecture of the convolutional neural network. The input of the model is a resize image with three channels (i.e. R, G, B). The CNN model has three convolutional and two layers of fully connected. The output of the model is a ten dimension vector corresponding to the estimated categories of the flowers.

4 Result and discussion

To verify the effectiveness of our model, we assessed its performance on validation sets. The accopanying Figure 2, Illustrate the confusion matrix for given validation dataset. We can observe that our model could not achieve the optimal performance across all 10 database classes (fleur0, fleur1....fleur8, fleur9) with recognition rate smaller than 30%(fleurs4). Furthermore, the highest recognition rate is 35% (7 on 20 images) which yielded an incorrect result for the classification between fleurs1 and fleurs2 due to the inherent dataset characteristics. Our model performed poorly in accurately categorizing the dataset. achieving only a average recognition rate of 10.5%.

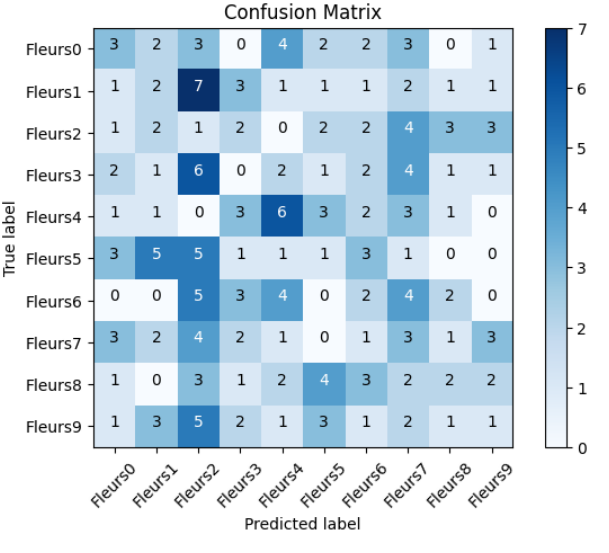

Figure 2: Confusion matrix of CNN method

As we can see above (Figure 2), the percentage of correct answers was low. Therefore, it is necessary to survey the model's accuracy. The Figure 3 showing the model's accuracy on the training and validation datasets. We can observe that when the training accuracy continues to improve while the validation accuracy start to decrease and constant. This suggest a sign of overfitting, which means the model is learning to memorize the traing data rather than generalizing well to new, unseen data. As introduced in the first chapter, to solve this kind of poor performance problem we would use the regurarization tools, data augmentation and Fine tuning a pre-trained CNN which is shown into the next chapter.

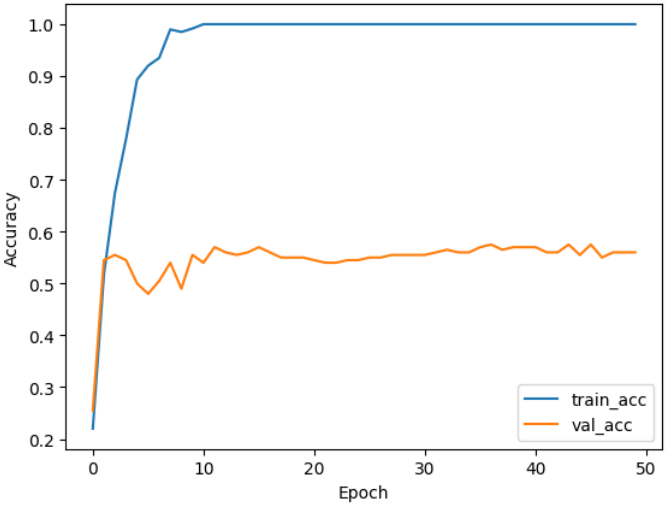

Figure 3: Training and validation curves

4.1 CNN + Data regularization

We implemented regularization in the model by applying two techniques such as L1/L2 regularization and Dropout. Testing the idea of incorporating regularization techniques like L1/L2 regularization and Dropout into a model is a good approach for several reasons [1]:

- Prevent overfitting: Regularization techniques help prevent overfitting, which occurs when a model performs exceptionally well on the training data but poorly on unseen data. By adding regularization, the model's complexity is reduced, making it more likely to generalize well to new data.
- Improved generalization: Regularization encourages the model to rely less on specific neurons or weights, making it more robust to variations in the input data. This results in a model that is more likely to perform well on unseen data.
- Better model performance: Regularization techniques can help the model achieve better performance by reducing the risk of overfitting and improving generalization. This is especially important when dealing with small datasets, as overfitting is more likely to occur in such situations.

In the process of implementation, the L1/L2 regularization technique was initially utilized by specifying the kernel_regularizer=regularizers.l2(0.01), whereby a penalty term was incorporated based on the L2 norm of the weight matrix. This regularization approach effectively curtails the complexity of the model and mitigates overfitting. Additionally, to improve the generalization performance of the model, the Dropout(0.5) technique was adopted, which involved random deactivation of 50% of the neurons during each epoch, thereby reducing the model's reliance on specific neurons.

4.2 CNN + Regularization + Data augmentation

Data augmentation is a method of generating new data by adding transformations to the original image or performing operations such as rotation, shifting and flipping. This can have the effect of increasing the size of the dataset, improving the accuracy of machine learning and preventing overfitting.

When performing data augmentation, it is necessary to consider whether the generated image is realistic. It should also be similar to the original image, so appropriate parameters need to be selected. In this case, the image angle and horizontal inversion were used as parameters for data augmentation. These parameters can be used to generate new data that are not far removed from the features of the original image.

4.3 CNN + Regularization + Data augmentation + Fine tuning a pre-trained

To performs fine-tuning we loading a pre-trained VGG16 model and then adding to the model that is already defined. After training, the last classification layer would be adapted to the new dataset, while re-training the features acquired by the pre-trained VGG16 models.

The VGG16 model is a convolutional neural network trained on the ImageNet dataset and designed to solve 1000 classes of image recognition tasks in the ImageNet dataset. VGG16 is known for its very high performance in image classification. It is known to perform well, and its very deep network structure makes it suitable for fine-tuning, as it can easily benefit from pre-training with huge datasets.

5 Method comparison

We conducted the implementation of each method and subsequently evaluated them on identical validation sets. The graphical representation of the outcomes is presented in Figure 4. Our findings indicate that with incremental improvements, the model's performance exhibited a steady increase. Notably, the Fine-tuning technique yielded the most superior performance among all the methods tested. The evaluation of the training model on the given test data is shown into the table 1.

To demonstrate the perfermance of above method we compared it with their test-set prediction. As shown in Figure 5, outperformed all the methods in comparison. The average recognition rate of each method are: CNN: 10.5%, CNN+Reg: 13%, CNN+Reg+DA: 11.5, CNN+Reg+DA+Fine tuning: 11.5% respectively.

Table 1: The accuracy of each method

Method	accuracy
CNN model	0.54
CNN + Reg	0.58
CNN + Reg + DA	0.64
CNN + Reg + DA + Fine-tuning	0.75

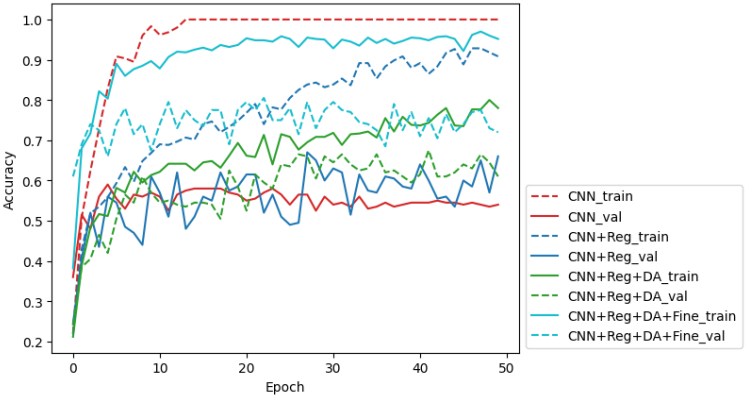

Figure 4: Training and validation curves of each method

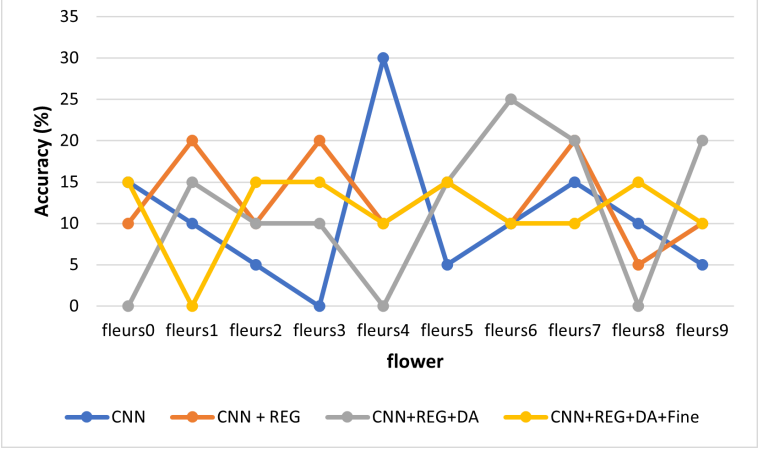

Figure 5: Performance comparision on the test set

6 Conclusion

The tasks in this exercise used a deeper architecture with few layers and small data, which cause the model to poorly capture more complex features and patterns in the images. However, we could explore various techniques such as data augmentation and regularization, which improved the robustness and generalization of the model into some level. Additionally, It would be necessary to increase to dataset and the layers of convolutional to obtain good result in the future.
The source code and datasets are available here. $https://github.com/tut203383/Flower\_project$.

References

[1] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.