CNN Classifier

RE6124019
Ming-Hsuan Wu 吳明軒
Institute of Data Science
RE6124019@gs.ncku.edu.tw

Abstract

In this study, the implementation of a convolutional neural network (CNN) classification model for animal image recognition and classification was applied to a dataset comprising images of three different animals: dogs, cats and pandas. Three distinct types of models were employed for the purpose of training. Initially, the AlexNet model was utilised without any form of enhancement, resulting in a test accuracy of approximately 0.72. Secondly, three improvement strategies were employed. Data augmentation techniques were employed, including rotation, horizontal flip and colour adjustment, with the objective of enhancing feature diversity. A bespoke ResNet50 was constructed with the objective of increasing the depth of the network, thereby improving overall performance. The learning rate is reduced in a gradual manner, which facilitates the training of the model and promotes convergence. The final accuracy achieved was approximately 0.86. The ResNet50 model provided by Pytorch was employed for training, including pretrained weights. This resulted in an accuracy of approximately 0.98.

1. Introduction

In recent years, convolutional neural networks (CNNs) [4] have made a significant contribution to the advancement of image recognition and classification techniques. This paper presents the development and implementation of a convolutional neural network (CNN)-based model for the classification of animal images. The paper examines a number of different architectures, commencing with AlexNet and progressing to a more sophisticated ResNet-50 model. The objective is to achieve high accuracy in classifying images of three different animal categories: dogs, cats, and pandas 1.

2. Related Work

This section provides an overview of related work in the field of convolutional neural networks (CNNs), with a particular focus on two significant architectures: The architectures in question are AlexNet and ResNet. These architectures have been instrumental in advancing the state of the art in image classification tasks [5].

2.1. AlexNet

AlexNet, proposed by Krizhevsky et al [3]. in 2012, marked a breakthrough in the field of deep learning and computer vision. It achieved a top-5 error rate of 15.3% in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012, significantly outperforming the previous state of the art. AlexNet consists of five convolutional layers followed by three fully connected layers 2. Key innovations introduced by AlexNet include the use of ReLU activation functions [1], dropout for regularization, and data augmentation techniques such as random cropping and horizontal flipping. The success of AlexNet demonstrated the potential of deep CNNs for large-scale image classification tasks.

2.2. ResNet

ResNet, introduced by He et al [2]. in 2015, further advanced the field of deep learning by addressing the vanishing gradient problem that often occurs in very deep networks. ResNet employs a novel architecture with ”residual blocks,” where shortcut connections allow gradients to bypass one or more layers, enabling the training of networks with over a hundred layers 3. This architecture led to significant improvements in image classification performance, achieving a top-5 error rate of 3.57% in the ILSVRC 2015 competition [5]. The ResNet family includes several variants such as ResNet-50, ResNet-101, and ResNet-152, named according to the number of layers. In our work, we leverage the ResNet-50 model due to its balance between depth and computational efficiency.

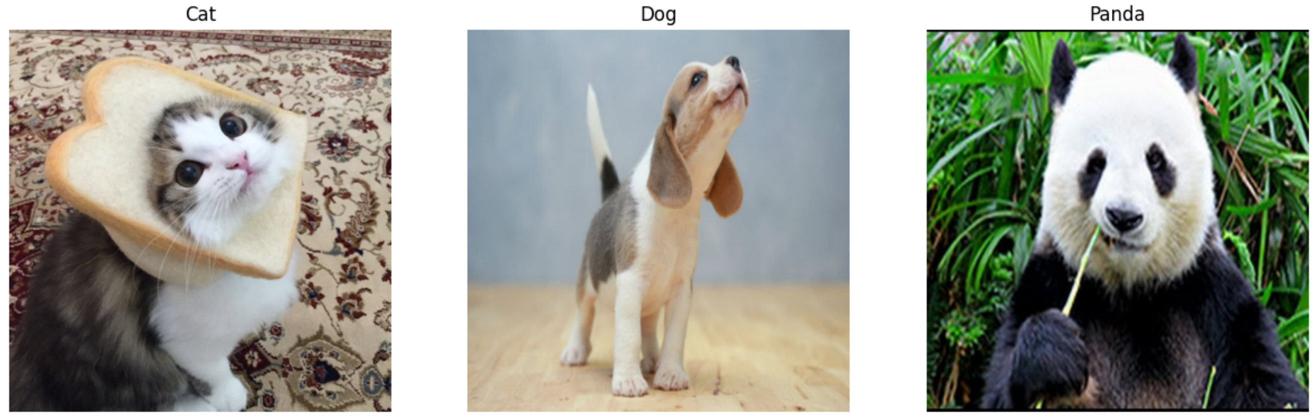

Figure 1. Example images from the dataset after resizing use `transform`. The dataset includes three categories of animals.

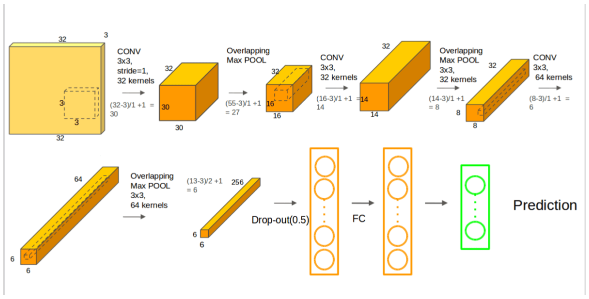

Figure 2. Architecture of AlexNet.

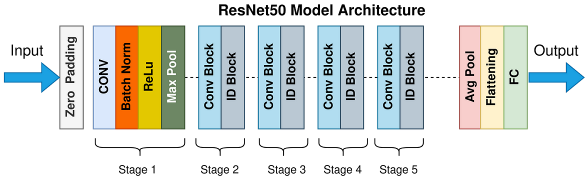

Figure 3. Architecture of ResNet-50.

In this study, we initially employed the AlexNet model and subsequently improved our model by transitioning to the ResNet-50 model. The pre-trained ResNet-50 model, which was fine-tuned on our specific dataset, demonstrated superior performance, indicating the benefits of using deeper and more advanced architectures for complex image classification tasks.

3. Proposed Method

This section presents a comprehensive account of the experimental process, which is divided into four main parts: loading packages, preparing data, building the model, and evaluation. Each stage of the process is described in detail to ensure the reproducibility and clarity of the experimental workflow.

3.1. Loading Packages

In the initial phase of the project, all necessary packages were imported, including those commonly used for data processing and visualization, such as `numpy`, `matplotlib.pyplot`, and `seaborn`. Additionally, PyTorch-related packages were included. Furthermore, a random seed (`SEED = 0`) was set to guarantee the reproducibility of the experiments. This ensures that each run of the script yields the same results. The importation of these packages establishes the requisite environment for subsequent data processing, model construction, training, and performance evaluation, thereby ensuring the generation of accurate and reliable results.

3.2. Preparing Data

(i) Importing Data: The `datasets.ImageFolder` method from PyTorch was used to prepare the dataset. This method efficiently loaded the image data and created a dataset with appropriate labels, allowing easy manipulation and processing for subsequent tasks.

(ii) Splitting Data: Once the dataset had been prepared, it was randomly divided into three distinct sets: a training set, a validation set, and a test set. The dataset was divided into three distinct subsets, each comprising a different proportion of the total data. The first subset, comprising 80% of the data, was allocated for training purposes. The second subset, comprising 10% of the data, was allocated for validation purposes. The third subset, comprising 10% of the data, was allocated for testing purposes. The aforementioned split resulted in 2,400 images being allocated for training, 300 images for validation, and 300 images for testing. This ensured that each subset contained a sufficient amount of data for robust model training and evaluation.

(iii) Data Augmentation: The objective of data augmentation was to enhance the diversity of the training dataset, which is a crucial factor in improving the model's generalisation ability. Initially, the images were simply resized to 224×224 pixels without any augmentation. For the subsequent model training, more complex augmentations were applied using `transforms`. These included random cropping and resizing, horizontal and vertical flips, random perspective transforms, and rotations between -15 to 15 degrees.

(iv) Data Loading: In order to facilitate the efficient training of the model, the dataset was converted into batches using PyTorch's `DataLoader`. The data was divided into minibatches of 100 images each and shuffled to ensure that the training process was random. This configuration permitted the effective utilisation of the dataset and facilitated enhanced training efficiency through optimisation of memory usage.

3.3. Building the Model

This section defines the model architecture and the optimization process. Initially, AlexNet was employed as the base model for the classification task. However, to improve performance, a more advanced architecture, ResNet50, was transitioned to. Ultimately, PyTorch's pre-trained ResNet50 model was utilized, with its final layers modified to fit the specific classification task. Furthermore, the loss function and optimizer were set up.

(i) Initial Model: The investigation commenced with AlexNet, a straightforward and extensively utilized convolutional neural network architecture. Despite its popularity, AlexNet's performance on our dataset was suboptimal, prompting us to explore more sophisticated models.

(ii) Improved Model: In order to enhance the model's performance, we adopted ResNet50, a deeper and more powerful architecture. The ResNet50 architecture, with its residual learning framework, addresses the vanishing gradient problem and enables the training of much deeper networks.

(iii) Final Model: PyTorch's pre-trained ResNet50 model, which is equipped with weights that have been trained on a substantial benchmark dataset (ImageNet), was employed in this study. This approach provided a robust foundation, enabling the utilisation of transfer learning through the fine-tuning of the network on the specific dataset. The final fully connected layer was modified to correspond to the number of classes in the task. Furthermore, the loss function was defined as Cross-Entropy Loss, and the Adam optimiser was employed for training.

3.4. Evaluation

Finally, the model's performance is evaluated using the test dataset. Metrics such as accuracy and F1 score are calculated in order to determine the extent to which the model is able to generalise to new, unseen data. Furthermore, the confusion matrix (CM) and the loss and accuracy curves are plotted during the training process.

(i) Calculating Accuracy and F1 Score: In order to assess the model's performance, we initially calculated the accuracy by comparing the model's predictions with the true labels. In order to provide a more comprehensive evaluation, particularly in the presence of class imbalance, the F1 score, which considers both precision and recall, was also computed.

(ii) Plotting Confusion Matrix: A confusion matrix was constructed in order to provide a visual representation of the performance of the classification model in distinguishing between different classes. The confusion matrix facilitated the identification of specific classes where the model exhibited suboptimal performance.

(iii) Plotting Loss and Accuracy Curves: During the training process, we monitored the loss and accuracy for both the training and validation sets. By plotting these curves, it was possible to gain insight into the model's learning progress and to identify any indications of overfitting or underfitting.

4. Experimental Results

This section presents the results of the experiments conducted, which have been divided into three main parts: model performance evaluation, confusion matrix analysis, and training process visualization.

4.1. Model Performance Evaluation

To assess the effectiveness of our models, we calculated the accuracy and F1 score on the test dataset. These metrics provide a comprehensive understanding of each model's ability to generalise to unseen data, highlighting their strengths and weaknesses. The results, presented in table 1, provide a clear comparison of the performance of different models.

Model	Accuracy	F1 Score
AlexNet	0.7167	0.7199
ResNet50 (Improved)	0.8600	0.8603
ResNet50 (Pre-trained)	0.9767	0.9767

Table 1. Performance evaluation of different models on the test dataset.

The pre-trained ResNet50 model demonstrated superior performance compared to the other models, achieving an accuracy of 0.9767 and an F1 score of 0.9767. This evidence serves to illustrate the efficacy of utilising deeper and pre-trained networks for the classification of complex images.

4.2. Confusion Matrix Analysis

To gain further insight into classification performance, we plotted the confusion matrix for each model. This provides an insight into each model's ability to categorise animal types and identify misclassifications.

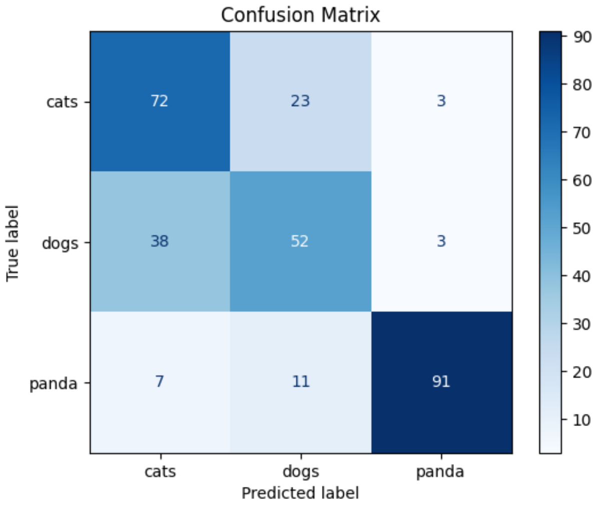

Figure 4. Confusion Matrix for the AlexNet model.

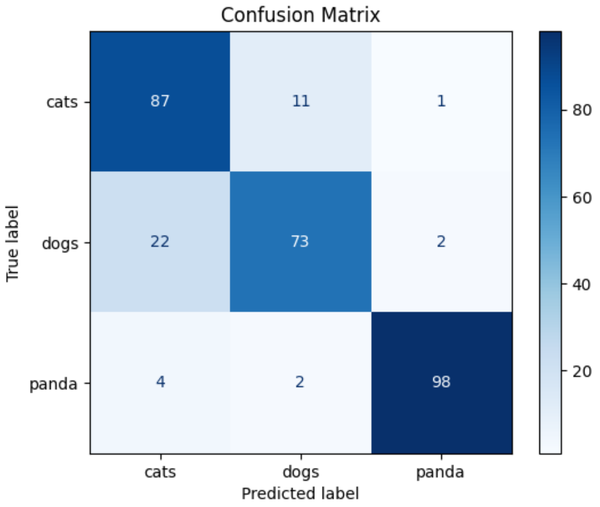

Figure 5. Confusion Matrix for the ResNet50 (Improved) model.

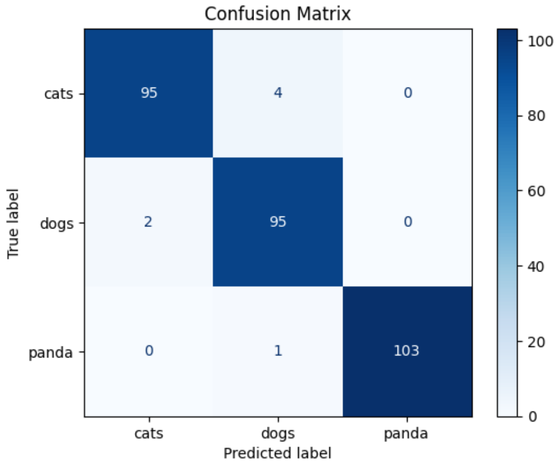

Figure 6. Confusion Matrix for the ResNet50 (Pre-trained) model.

The confusion matrices in Figures 4, 5 and 6 provide a comparative analysis of the classification performance of the AlexNet, improved ResNet50 and pre-trained ResNet50 models respectively. The AlexNet model 4 shows significant misclassifications, particularly between cats and dogs, indicating limited generalisation. The improved ResNet50 model 5 shows improved performance but still has some misclassifications, showing better but not perfect discrimination between classes. The pre-trained ResNet50 model 6 shows high precision and recall across all categories, with minimal misclassifications, indicating that it is highly effective in distinguishing between the different animal classes.

4.3. Training Process Visualization

In order to provide a visual representation of the training process, we have plotted the loss and accuracy curves for both the training and validation sets. The aforementioned curves offer insights into the model's learning progress and facilitate the detection of any indications of overfitting or underfitting.

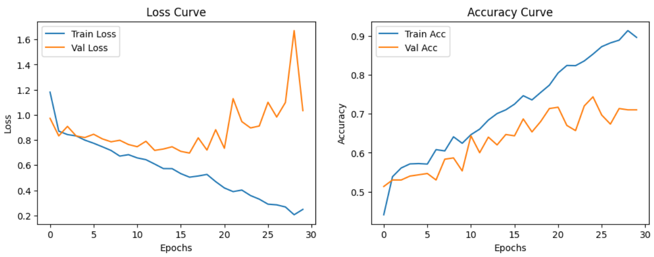

Figure 7. Training and Validation Loss and Accuracy for AlexNet

Figure 7 shows the training and validation loss and accuracy curves for the AlexNet model. The training loss gradually decreases while the validation loss remains high and variable, indicating overfitting. Similarly, the training accuracy increases significantly compared to the fluctuating validation accuracy, further suggesting overfitting.

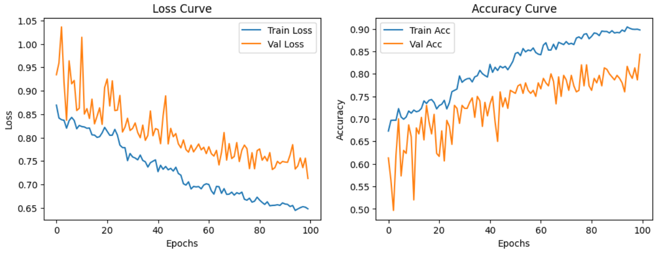

Figure 8. Training and Validation Loss and Accuracy for ResNet50 (Improved)

Figure 8 shows the training and validation loss and accuracy curves for the improved ResNet50 model. Both losses decrease over time, indicating continuous learning and improvement even after 100 epochs. The training accuracy increases steadily, indicating that the model does not overfit and generalises well to the validation data.

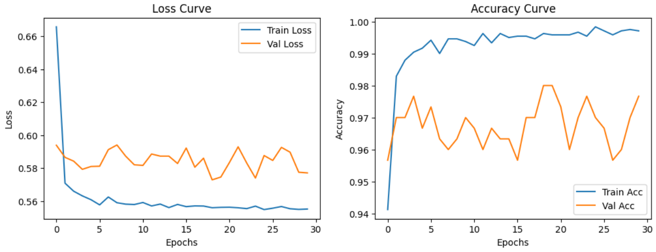

Figure 9. Training and Validation Loss and Accuracy for ResNet50 (Pre-trained)

Figure 9 shows the training and validation loss and accuracy curves for the pre-trained ResNet50 model. The model converges quickly, reaching high accuracy within the first few epochs. Although there is slight overfitting, the validation loss and accuracy remain close to the training metrics, indicating good generalization. Due to this rapid convergence and strong performance, the model was trained for only 30 epochs.

The training curves indicate that the pre-trained ResNet50 model achieved a stable learning process with no significant overfitting, as the validation metrics closely followed the training metrics throughout the training epochs.

5. Conclusion

In conclusion, the results of this study indicate that the implementation of convolutional neural network (CNN) models for the classification of animal images, specifically dogs, cats, and pandas, is a promising avenue for further research. In conclusion, this paper presents a comprehensive study on the implementation of convolutional neural network (CNN) models for classifying animal images, specifically dogs, cats, and pandas. The experiments demonstrated that the AlexNet model, despite its relatively shallow architecture and lack of enhancements, achieved an accuracy of approximately 75%, indicating that even basic CNN models can perform reasonably well. The enhanced ResNet50 model, which incorporates various improvement strategies, achieved an accuracy of approximately 85%, thereby demonstrating the benefits of model depth and enhancement techniques. Finally, the pre-trained ResNet50 model, trained on a large-scale dataset, demonstrated excellent performance, achieving nearly 100% accuracy. This highlights the advantage of utilising pre-trained models for achieving superior results in specific applications. In conclusion, the findings of this study support the value of both model depth and pre-training in achieving high accuracy in image classification. Further enhancements could be achieved through the experimentation of other advanced architectures, the utilisation of larger and more diverse datasets, and the incorporation of additional techniques to mitigate overfitting.

References

[1] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018. 1

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 1

[4] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 33(12):6999–7019, 2021. 1

[5] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 1