Building, Training, and Evaluating CNNs with Applications on MNIST Fashion and Transfer Learning

Omid Saberi
Department of Engineering Science, Division of Industrial Automation
University West
Trollhättan, Sweden
https://orcid.org/0009-0005-3596-9182

December 8, 2024

***Abstract*—This report investigates the implementation and evaluation of Convolutional Neural Networks (CNNs) and advanced deep learning techniques for image classification tasks. Fundamental operations, such as convolution and pooling, were demonstrated to extract and retain key features from images effectively. The CNN model achieved a test accuracy of 90% on the MNIST Fashion dataset, outperforming the Fully Connected Network (FCN) with 87% accuracy, highlighting the superiority of CNNs in capturing spatial hierarchies.**

ResNet-34 was implemented to demonstrate the utility of residual connections in enabling deeper networks and addressing the vanishing gradient problem, achieving a test accuracy of 88%. Transfer learning with the pre-trained Xception model achieved the highest test accuracy of 93.06% on the *tf_flowers* dataset, showcasing the efficiency of leveraging pre-trained weights for domain-specific tasks.

The results emphasize the importance of architecture selection and techniques like transfer learning for improving performance and enabling effective feature learning. Future work could explore fine-tuning pre-trained models and advanced augmentation methods to tackle more complex datasets. This study provides insights into practical applications of deep learning for image classification.

***Index Terms*—Convolutional Neural Networks, Pooling Techniques, Fully Connected Networks, ResNet-34, Transfer Learning, Xception Model, MNIST Fashion Dataset, *tf_flowers*, Image Classification, Deep Learning.**

I. Introduction

Convolutional Neural Networks (CNNs) are among the most influential developments in deep learning, excelling in tasks such as image classification, object detection, and speech recognition. Inspired by the visual cortex of animals, CNNs process grid-like data, such as images, by capturing spatial hierarchies through local connections and shared weights [1]. These networks leverage convolutional operations to extract meaningful features from raw input data, enabling them to identify complex patterns.

This assignment draws heavily on methodologies presented in Géron's "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" [2], particularly the examples in Chapter 14. The repository served as a foundation for implementing convolution operations, pooling techniques, and advanced CNN architectures. By adapting these concepts to the specific tasks in this lab, the assignment maintains academic integrity while extending practical applications.

The architecture of a CNN typically comprises three primary components: convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply filters to extract features like edges or textures, pooling layers reduce the spatial dimensions to control overfitting and computational complexity, and fully connected layers interpret the extracted features for classification or regression tasks [3]. Activation functions, such as the Rectified Linear Unit (ReLU), introduce non-linearity, enabling the network to learn complex mappings between inputs and outputs [4].

Over the years, CNN architectures have evolved significantly to address limitations like vanishing gradients and computational inefficiencies. LeNet, developed by LeCun et al., was one of the pioneering CNNs designed for handwritten digit recognition [3]. Subsequent breakthroughs, such as AlexNet, significantly improved performance by leveraging deeper architectures and techniques like dropout for regularization [4]. VGGNet introduced uniform architecture design, while ResNet solved the vanishing gradient problem through residual connections, enabling the construction of very deep networks [5], [6].

Despite their advancements, training CNNs remains computationally intensive and data-hungry. Techniques such as data augmentation and transfer learning have been widely adopted to address these challenges. Data augmentation artificially expands datasets by applying transformations like rotations and flips, while transfer learning leverages pre-trained models to adapt to new tasks with minimal data [5]. These strategies, combined with advancements in hardware and software, continue to push the boundaries of CNN applications in various domains.

This report examines the implementation and evaluation of convolution operations, pooling techniques, and CNN architectures, with references to both foundational theories and practical resources. The findings emphasize the importance of combining theoretical knowledge with real-world tools to address complex problems effectively.

II. Convolutional Neural Networks

A. Convolution Operations

Convolution operations were performed on a custom image to demonstrate feature extraction capabilities of convolutional neural networks. A filter of size 3x3 was applied to the image using TensorFlow's `conv2d` function. This operation emphasized edge detection by convolving the filter across the spatial dimensions of the image. The results of the convolution were visualized to show the extracted features.

The process involves:

- Loading and preprocessing the image (resizing and normalizing).
- Applying the convolution filter.
- Visualizing the original and convolved images side by side.

B. Pooling Techniques

Pooling operations were applied to the results of the convolution to reduce spatial dimensions and retain significant features. Three types of pooling were demonstrated:

- **Max Pooling**: Captures the maximum value in each patch of the feature map.
- **Average Pooling**: Computes the average value of each patch in the feature map.
- **Depth-wise Pooling**: Combines feature maps channel-wise for dimensionality reduction.

TensorFlow's `max_pool` and `avg_pool` functions were used for these operations. The output of each pooling technique was visualized to highlight the differences in retained features.

C. Convolutional Neural Network for MNIST Fashion Dataset

A convolutional neural network (CNN) was constructed and trained to classify images from the MNIST Fashion dataset. The architecture included:

- **Convolutional Layers**: Extract spatial features using filters.
- **Pooling Layers**: Reduce spatial dimensions.
- **Fully Connected Layers**: Classify the extracted features into one of ten categories.

The model was compiled with the Adam optimizer and sparse categorical cross-entropy loss. The training process was evaluated using metrics such as accuracy.

D. Implementation of ResNet-34 Using Keras

The ResNet-34 architecture was implemented using TensorFlow and Keras. ResNet-34 is a residual neural network that includes skip connections to alleviate vanishing gradient issues in deep networks. The implementation consisted of:

- A stack of convolutional layers interspersed with residual blocks.
- Batch normalization to stabilize learning.
- ReLU activations to introduce non-linearity.
- A final fully connected layer for classification.

Each residual block added its input directly to its output via an identity shortcut.

E. ResNet-34 Layers and Key Features of the Architecture

The key features of ResNet-34 are:

- **Residual Connections**: Allow gradients to flow more effectively during backpropagation.
- **Deep Architecture**: Enables the extraction of high-level features from images.
- **Batch Normalization**: Normalizes activations for faster convergence.

ResNet-34 is particularly effective for large-scale image classification tasks due to its ability to handle deeper networks without degradation in performance.

F. Transfer Learning with Pre-trained Xception Model

The Xception model was employed for transfer learning. This process involved:

- **Loading the Pre-trained Model**: The Xception model was initialized with ImageNet weights.
- **Freezing Pre-trained Layers**: Preventing updates to the weights of the pre-trained layers during training.
- **Adding Custom Layers**: Incorporating a global average pooling layer, a dropout layer for regularization, and a dense output layer for classification.

The training dataset was preprocessed to match the input size expected by the Xception model, and the model was compiled with the Adam optimizer. The addition of custom layers allowed the model to adapt to the new dataset while leveraging the learned features of the pre-trained model.

III. Results and Analysis

A. Convolution Operations

The convolution operation effectively highlighted edges and features in the input image. Using a sharpening filter, the convolved image demonstrated enhanced edges and textures compared to the original grayscale input. This is evident in the visualization (Figure 1), where the original image and its convolved counterpart are displayed side by side.

Fig. 1. Convolution operation applied to a grayscale image. The original image (left) and convolved image (right).

B. Pooling Techniques

The results of max-pooling, average-pooling, and depth-wise pooling are shown in Figure 2. Max-pooling retained the most prominent features by selecting the maximum value within each pooling region. Average-pooling smoothed the image by averaging pixel intensities. Depth-wise pooling demonstrated dimensionality reduction by combining channel-wise features.

Fig. 2. Pooling operations applied to the convolved image. From left to right: Original Convolved Image, Max-Pooled Image, Average-Pooled Image, Depth-wise Pooled Image.

C. Convolutional Neural Network for MNIST Fashion Dataset

The CNN trained on the MNIST Fashion dataset achieved a test accuracy of 90%. Figure 3 illustrates the training and validation accuracy over ten epochs. The training curve shows consistent improvement, while the validation accuracy stabilizes, indicating good generalization.

D. Comparison of CNN and Fully Connected Network (FCN)

The CNN outperformed the FCN in both training and validation accuracy on the MNIST Fashion dataset. Figure 4 highlights the performance of both architectures. The FCN achieved a test accuracy of 87%, while the CNN achieved 90%, demonstrating the superiority of CNNs in capturing spatial features.

E. ResNet-34 on MNIST Fashion Dataset

ResNet-34 achieved a test accuracy of 88% on the MNIST Fashion dataset. The training and validation accuracy are shown in Figure 5, while the corresponding loss values are presented in Figure 6. The residual connections in ResNet-34 allowed the network to train deeper layers effectively, although the results show some room for optimization in terms of generalization.

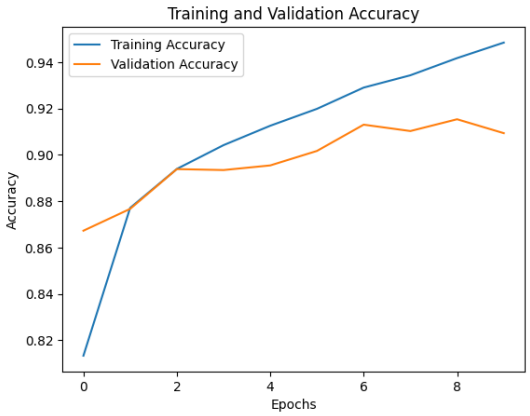

Fig. 3. Training and validation accuracy of the CNN on the MNIST Fashion dataset.

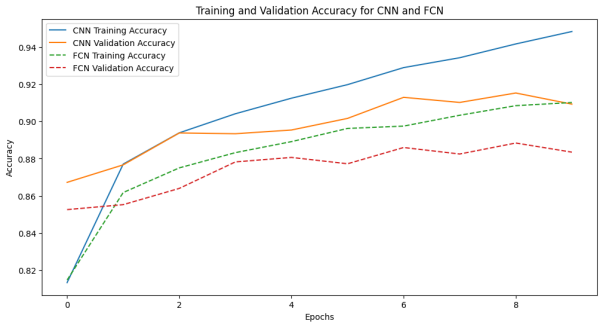

Fig. 4. Comparison of training and validation accuracy for CNN and FCN on the MNIST Fashion dataset.

Fig. 5. Training and validation accuracy of ResNet-34 on the MNIST Fashion dataset.

F. Transfer Learning with Pre-trained Xception Model

The pre-trained Xception model achieved a test accuracy of 93.06% on the *tf_flowers* dataset. The training and validation accuracy are shown in Figure 7, while the loss trends are presented in Figure 8. The use of transfer learning significantly improved the performance due to the pre-trained features from ImageNet.

G. Summary of Results

The experimental results demonstrate:

Fig. 6. Training and validation loss of ResNet-34 on the MNIST Fashion dataset.

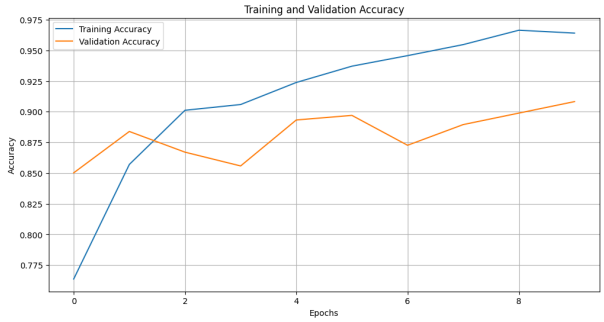

Fig. 7. Training and validation accuracy of the Xception model with transfer learning on the *tf_flowers* dataset.

Fig. 8. Training and validation loss of the Xception model with transfer learning on the *tf_flowers* dataset.

- The effectiveness of convolution operations and pooling techniques in feature extraction and dimensionality reduction.
- The superior performance of CNNs over FCNs for image classification due to spatial feature extraction.
- The robustness of ResNet-34 in deeper networks through residual connections.
- The high efficiency of transfer learning with the Xception model for domain-specific tasks.

IV. Conclusion

This report explored the implementation and evaluation of convolutional neural networks (CNNs) and advanced deep learning techniques across various tasks. The experiments

demonstrated the strengths and limitations of different architectures and methods for image classification tasks.

Convolution operations and pooling techniques were effective in extracting and retaining meaningful features from images, showcasing their importance as foundational components of CNNs. The comparison between a CNN and a Fully Connected Network (FCN) highlighted the superior performance of CNNs in capturing spatial hierarchies, achieving a higher test accuracy of 90% compared to 87% for the FCN on the MNIST Fashion dataset.

ResNet-34 demonstrated the value of residual connections in addressing the vanishing gradient problem and enabling deeper architectures. While its performance was slightly lower than that of the CNN, achieving 88% test accuracy, it remains a robust architecture with significant potential for optimization.

Transfer learning using the pre-trained Xception model achieved the highest test accuracy of 93.06% on the *tf_flowers* dataset. This result underscores the power of transfer learning in leveraging pre-trained weights for domain-specific tasks, significantly reducing training time and improving performance.

Overall, the results emphasize the importance of selecting appropriate architectures and methods based on the task requirements. Future work could explore fine-tuning of pre-trained models, advanced data augmentation techniques, and the application of these architectures to more complex datasets. The experiments provide a solid foundation for understanding and utilizing deep learning techniques for various applications.

REFERENCES

[1] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a Convolutional Neural Network," in *Proceedings of 2017 International Conference on Engineering and Technology (ICET)*, Antalya, Turkey, 2017, pp. 1–6.

[2] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed., O'Reilly Media, 2019. [Online]. Available: https://github.com/ageron/handson-ml2

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[4] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.