Comparing Transfer Learning Performance of Convolutional Neural Networks and Vision Transformers

Kelvin Kuang

1. Introduction

Prior to the introduction of attention in 2017[1] and the vision transformer (ViT) in 2020[2], convolutional neural networks (CNNs) were deemed the gold standard for computer vision and image recognition tasks. CNNs, through their deep layers of convolutional filters, were proficient at extracting features from images, leading to significant advances in various applications such as object detection, facial recognition, and medical imaging. However, the emergence of the attention layer and transformer architecture marked a paradigm shift. The transformer architecture, initially popularized in natural language processing and large language models (LLMs) such as ChatGPT, offered a new approach to handling sequential data, focusing on the most relevant parts of the input through context vectors. This concept was further evolved with the introduction of ViTs, which adapted the transformer architecture for image processing. ViTs demonstrated an exceptional ability to understand global contexts within images, challenging the supremacy of CNNs in large-scale image recognition tasks. This shift not only revolutionized the field of computer vision but also opened avenues for hybrid models, combining the strengths of both CNNs and transformers. The 2020 study *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale* by Dosovitskiy et al.[2] concluded that "vision transformers attain excellent results compared to state-of-the-art convolutional networks while requiring substantially fewer computational resources to train." In this study, I attempt to recreate this experiment on a smaller scale with a smaller dataset compared to the ImageNet dataset.

2. Methodology

2.1. Data Preparation

For this study, I fine-tuned CNN and ViT models on a card images dataset from Kaggle[3], consisting of 53 classes, each class consisting of a cards suit and rank. Each class contained approximately 130 images for training, 5 for validation, and 5 for testing, creating a 80:20:20 split for training, validation, and testing. This dataset was considerably neat and tidy, with properly labeled images, little to none background noise, and a uniform 224x224x3 size for training. Horizontal and vertical flip augmentations were performed on the images to add more variation in the data.

2.2. Fine-Tuning Convolutional Neural Networks (CNNs)

In the study by Dosovitskiy et al.[2] ResNet50, ResNet101, and ResNet152 were the chosen CNN architectures. In this study, due to the smaller dataset and downsizing of the study, ResNet50 was the chosen architecture. The top layer was replaced with a dense layer with 53 units (corresponding to the 53 classes) with a softmax activation function. Dropout was introduced with p=0.4 before the dense layer, and all previous layers were frozen. Cross-entropy was used as the loss function. To optimize performance, a random search determined the most effective learning rate. The training and architecture was implemented using Keras, and was conducted over 20 epochs for time and computational resource considerations.

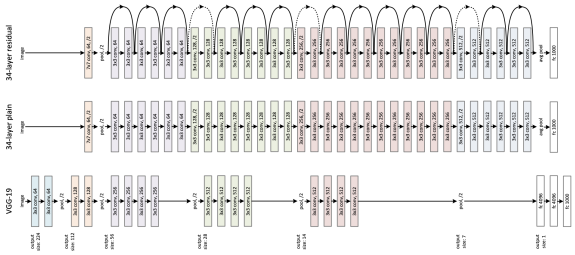

Figure 1: Architecture of a residual neural network, aka ResNet (top), compared to a linear network (middle) and VGG-19 (bottom). Figure obtained from *Deep Residual Learning for Image Recognition* by He et al.[4] The ResNet architecture differs from a plain linear network through the inclusion of skip connections, which perform an identity mapping, where the output of a layer is added to another layer deeper in the network.

2.3. Fine-Tuning Vision Transformers (ViT)

Again, in the study by Dosovitskiy et al.[2], ViT-B/{16,32}, ViT-L/32, ViT-L/16, and ViT-H/14 were evaluated. For this smaller-scale study, ViT-B/16 was the chosen architecture. The transformer architecture was implemented through PyPI, and training through PyTorch. The loss function was also cross-entropy. Training was limited to 2 epochs with the Adam optimizer and a learning rate of 1e-4 for time and computational resources consideration.

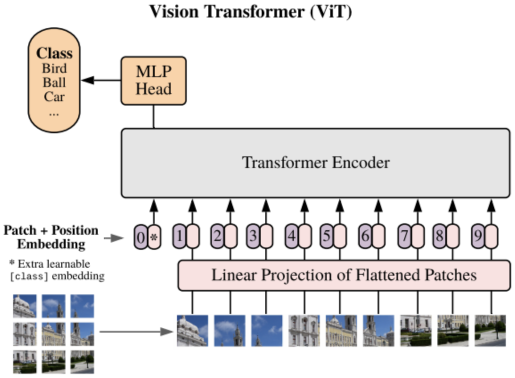
Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0 *
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Figure 2: Architecture of a vision transformer (ViT). Figure obtained from Dosovitskiy et al.[2] . A transformer block contains 2 residual blocks, with each residual block having a skip connection. The first residual block is a multihead attention block, which performs self-attention and computes a context vector. The second residual block is a feed-forward layer that applies a nonlinear transformation to the output of the self-attention layer. The multihead attention also maintains positional embeddings.

3 Results

3.1. CNN Transfer Learning Results

Fine-tuning of ResNet50 saw considerable overfitting, with the final training loss and accuracy resulting in 0.4765 and 0.8534 respectively, but the validation loss and accuracy was considerably higher: 1.7081 and 0.5434 respectively. Given that there are

53 classes, this model performed better than chance level accuracy (which would be 0.01887, 1/53), but performed considerably worse than the ViT. Even though the earlier layers were frozen to decrease the number of trainable parameters, overfitting is not surprising given the amount of training data, and the spare image augmentation. In the continuation of this study, more data augmentation will be used to increase the variability of the dataset, and L2 regularization will also be considered to combat overfitting.

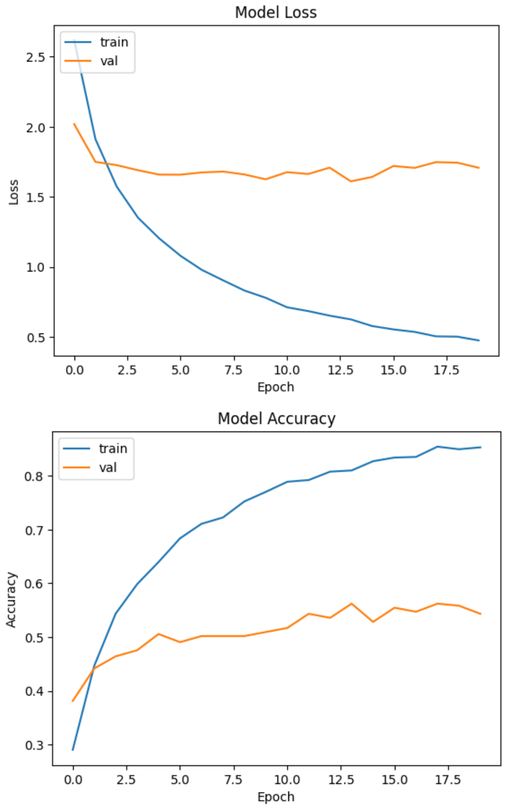

Figure 3: Training and validation curves for loss and accuracy of ResNet50 fine-tuning. Starting around the 2nd epoch, the model starts to overfit. The training loss continues to decrease, but the validation loss starts to plateau. The same trend is seen in the accuracy plot. This is obvious signs of overfitting, which could be combated by implementing L2 regularization or adding more data augmentations. This will be taken into consideration for the continuation of this project.

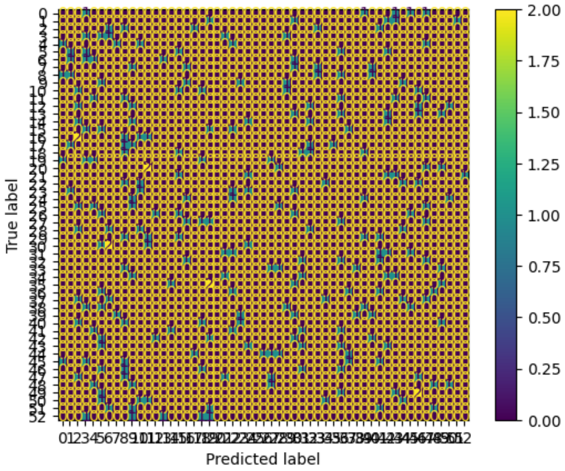

Figure 4: Confusion matrix of CNN on testing set. Labels are 0-52, each corresponding to a class. Out of the 265 testing images, only 2 were classified correctly, resulting in a 0.007547 accuracy. This accuracy is worse than chance level accuracy, and suggests that the CNN was not able to learn any key features to be able to distinguish between different classes.

3.2. ViT Transfer Learning Results

Compared to the CNN transfer-learning performance, the ViT did significantly better with 90% less epochs (2 epochs for the ViT compared to 20 epochs for the CNN). The final validation loss and accuracy obtained was 0.0065367 and 0.9358. On the testing set, the ViT obtained 1.0 accuracy. The same dataset with the same data augmentations were used. Given that both CNN and ViT models were pre-trained on the Imagenet dataset, these results reaffirm the results from Dosovitskiy et al.[2], but this could be due to lack of optimization and training for the CNN.

4. Conclusions

The results of this study suggests that ViT's outperform CNN's drastically, but I believe this may just be a result of not optimizing the CNN training. With more data augmentation, implementation of L2 regularization, and training for more epochs, I believe that the accuracy of the CNN can increase and narrow the gap. Compared to other results using the same dataset, accuracies in the high 0.9s were achieved, but training took at a minimum 200 epochs. CNNs also seem to overfit easier compared to ViTs, as the CNN began to overfit by the 2nd epoch, but the ViT did not. This again may be due to poor training of the CNN, but will be investigated in the future.

5. References

[1] A. Vaswani *et al.*, "Attention Is All You Need," *arXiv.org*, Jun. 12, 2017. Available: https://arxiv.org/abs/1706.03762

[2] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv:2010.11929 [cs]*, Oct. 2020.

[3] "Cards Image Dataset-Classification," *www.kaggle.com*. Available: https://www.kaggle.com/datasets/gpiosenka/cards-image-datasetclassification

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv.org*, Dec. 10, 2015. Available: https://arxiv.org/abs/1512.03385