CS587 - Neural Networks & Learning of Hierarchical Representation

Spring Semester 2024
Assignment 3

Papageridis Vasileios - 4710
csd4710@csd.uoc.gr

May 4, 2024

Introduction

This assignment aims to teach us how the process of transfer learning works, at first, and familiarize us with the Image-Specific Class Saliency Visualization.
In the first part of this assignment, *Transfer Learning*, we explore the application of a convolutional neural network, AlexNet, to a new domain. Originally designed for image classification tasks on the ImageNet dataset, AlexNet's capabilities are extended through transfer learning. Our task involves fine-tuning AlexNet on the WikiArt dataset, which comprises 4000 images representing 10 distinct artistic styles, including Baroque, Realism, and Expressionism and others. This approach leverages the pretrained network's ability to extract meaningful features from images, adapting it to recognize and classify art styles effectively.
In the second part, we employing saliency visualization techniques with a pre-trained VGG-16 model. This model has been trained on the ImageNet dataset too. Here, we aim to uncover the model's decision-making process by creating saliency maps that highlight the most influential regions within an image that contribute to the model's classification decisions.

Figure 1: Sample image from the WikiArt dataset.

First Part: Transfer Learning

Training Loss Plot

Figure 2: Training Loss Plot showing the change in loss across training epochs.

The training loss plot displays a general downward trend, indicating that the model is effectively learning and improving across iterations. Initially high, the loss decreases consistently, which reflects positive learning dynamics. However, there are visible fluctuations in the loss values between batches, suggesting that certain batches might be more challenging, or that the model's parameters are still undergoing significant adjustments.

Training Accuracy Plot

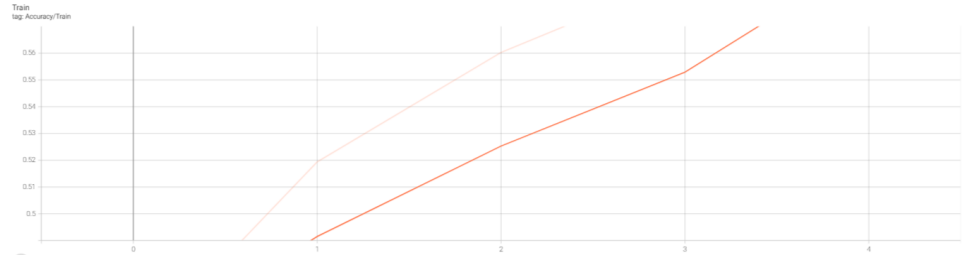

Figure 3: Training Accuracy Plot showing the increase in accuracy across epochs.

The training accuracy plot exhibits a clear upward trend over epochs, starting from slightly above 50% and climbing to approximately 66% by the end of the fifth epoch. This consistent improvement serves as a strong indicator of the model's capability to learn task-specific features effectively and to increasingly accurately classify the paintings into one of the ten art styles.

Test Accuracy Plot

The test accuracy plot reveals an initial increase in accuracy from the first to the third epoch, indicating that the model is beginning to generalize well to unseen data. However, there is a significant drop in the fourth epoch, suggesting potential overfitting to the training data, before a recovery in the fifth epoch. This pattern points to possible corrections made by the optimizer during training.

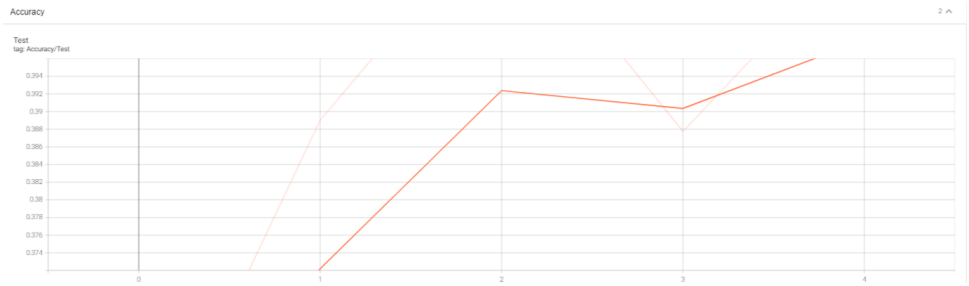

Figure 4: Test Accuracy Plot highlighting variability across epochs.

Observations

- **Overfitting Concerns:** The fluctuations in test accuracy, alongside higher training accuracy, suggest that the model may be overfitting to the training data. Implementing regularization techniques, enhancing data augmentation, could improve generalization.
- **Learning Rate and Batch Size:** Fine-tuning these parameters could help achieve more stable test accuracy and reduce observed fluctuations.
- **Model Capacity and Complexity:** Considering that AlexNet is a relatively older and simpler architecture, exploring newer models like ResNet, known for better generalization capabilities, might yield improved outcomes.

Analysis of the Results

1. **Training Accuracy Improvement:** There is a consistent increase in training accuracy from epoch 1 to epoch 5, starting at 44.8% and reaching 65.8%. This improvement indicates that the model is effectively learning from the training data across epochs.
2. **Test Accuracy Variability:** The test accuracy does not show a consistent improvement. It starts at 34.6% in the first epoch, peaks at 41.3% in the third epoch, and then decreases slightly in subsequent epochs. This variance and the peak followed by a decrease could indicate:
    - **Overfitting:** As the training accuracy increases but the test accuracy does not improve correspondingly, it suggests the model might be overfitting the training data.
    - **Learning Rate and Epochs:** The learning rate of 0.01 and training for only 5 epochs with a batch size of 128 might not be optimal.

Intuition Behind Removing Last or Last Two FC Layers

1. **Removing the Last Fully-Connected (FC) Layer:** The last FC layer in a pretrained model usually functions as the decision layer, mapping learned high-level features to specific class labels from the original ImageNet dataset. Replacing this layer with a new one that corresponds to the number of classes in the WikiArt dataset is going to adapt the model to a new classification task. This re-targeting allows the model to predict based on features learned from ImageNet but tuned to the context of the new data.

2. **Removing the Last Two FC Layers:** The penultimate layer generally acts as a feature aggregator, setting the stage for the final classification decision. Redefining this layer can significantly adapt the network to new data, enabling a reconfiguration of how abstract features are synthesized before classification. This approach may be beneficial for datasets like WikiArt, which, while visually analogous to the training data (ImageNet), differ substantially in content.

Justifying Overall Accuracy Based on Dataset Characteristics

1. **Dataset Size and Diversity:** The WikiArt dataset is smaller and less diverse in label variety compared to ImageNet, which might limit the diversity of visual features and scenarios presented during training. This can potentially constrain the model's ability to learn robust generalization across different art styles.

2. **Image Style and Complexity:** Art images differ from the natural scene photographs in ImageNet. Artistic styles incorporate abstract representations and unusual compositions, which might not be effectively captured by feature detectors optimized for more straightforward, natural images. This adds complexity to the transfer learning task, as the pretrained model may initially lack specialized feature detectors needed for optimal performance on art style classification.

3. **Transfer Learning Formulation:** The effectiveness of transferring knowledge from one domain (ImageNet) to another (WikiArt) largely depends on the relevance of the features learned in the source task to the target task. While high-level visual features (e.g., edges, textures) are likely transferable, more specific features (e.g., parts of objects) may not be relevant to the target task, potentially limiting overall accuracy even with fine-tuning.

Theoretical Questions Answers

Question: My dataset is small but similar to the original dataset. Should I fine-tune?
Answer: Yes, you should fine-tune. When the dataset is small but similar to the original dataset, fine-tuning a pre-trained model is beneficial. This approach takes advantage of the pre-trained features that are capable of capturing the necessary information from images similar to those in the original dataset. Fine-tuning allows these features to be slightly adjusted to better fit on a smaller dataset without the extensive training data required to train a model from scratch effectively.

Question: My dataset is large and similar to the original dataset. Should I fine-tune or train from scratch?
Answer: Both fine-tuning and training from scratch are good options when the dataset is large and similar to the original dataset. If computational resources and time are constraints, fine-tuning might still be preferable because it can achieve a good performance more quickly. However, training from scratch could also be beneficial since the large dataset provides enough data to effectively learn all the necessary features without relying on pre-trained weights. Training from scratch might lead to a model that is better optimized for the specific characteristics of the dataset.

Question: My dataset is different from the original. Should I fine-tune?
Answer: Fine-tuning can still be a good approach even if the dataset is different from the original, especially if there are some underlying similarities in the types of features that need to be recognized (e.g., both are image datasets). It's a good practice to replace more layers than typically would or train more layers of the network than just the classifier, as the differences in data characteristics might require deeper modifications to the learned features.

Second Part: Saliency Visualisation

In this part, we utilize a pre-trained VGG-16 model to compute class saliency maps, in order to visualize the areas in an image that significantly influence the classification decisions of a convolutional neural network. The VGG-16 model, trained on a subset of the ImageNet database, has learned rich feature representations for a wide array of images, making it a robust tool for deep visual insights.

Implementation Steps

The process to compute the saliency map involves those steps, as follows:

1. Perform a forward pass of the image through the network to obtain the raw class scores.
2. Compute the gradient of the scores with respect to the input image, focusing specifically on the score corresponding to the target class.
3. Modify the gradient such that only the gradient with respect to the target class is non-zero, effectively isolating the influence of each pixel on the target class's score.
4. Backpropagate these gradients through the network to generate the saliency map.
5. Render the computed gradients to visualize the saliency map, highlighting the areas most influential in the classification decision.

Results of classification

In this section we will discuss about the results of the classification, based on the model prediction and on the results of the saliency maps that we got as an output for each image. As we can clearly observe, the model classifies all the images correctly except the ”flamingo” image, as shown in figure 7. As the saliency map reveals, the sand of the image and the darker blue background on the right part of the image plays a big role in the classification of the image, which correspond to a seashore scene. The model can capture only the central body of the flamingo, but it is unable to capture other features of the animal, which is making it difficult to give emphasis on the object than the whole scene. Essentially, this is the reason why the image is misclassified as ”seashore” and not as ”flamingo”. Also, this suggests that the subset of ImageNet used in order to train the VGG-16 model may not include enough pictures of a flamingo in order to make the model capable to classify this kind of animal.

Figure 5: On the left: Original image of the cat. On the right: Saliency map for the image.

Figure 6: On the left: Original image of the doberman. On the right: Saliency map for the image.

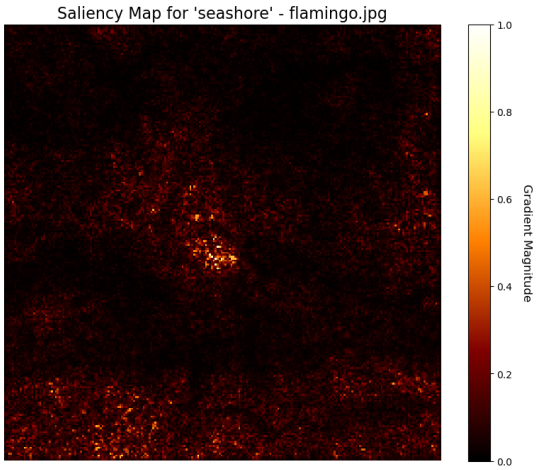

Figure 7: On the left: Original image of the flamingo. On the right: Saliency map for the image.

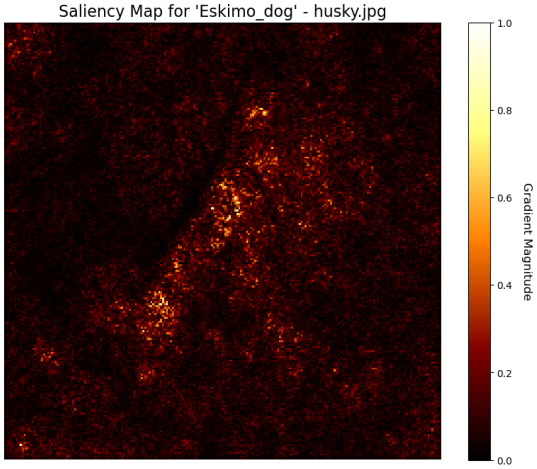

Figure 8: On the left: Original image of the husky. On the right: Saliency map for the image.

On the other side, we can clearly observe that each image, even some challenging ones, are correctly classified from the model. The saliency maps of all the other images, except the flamingo do highlight relevant parts of the images in order to recognize with a high accuracy the animals/objects of the given images.

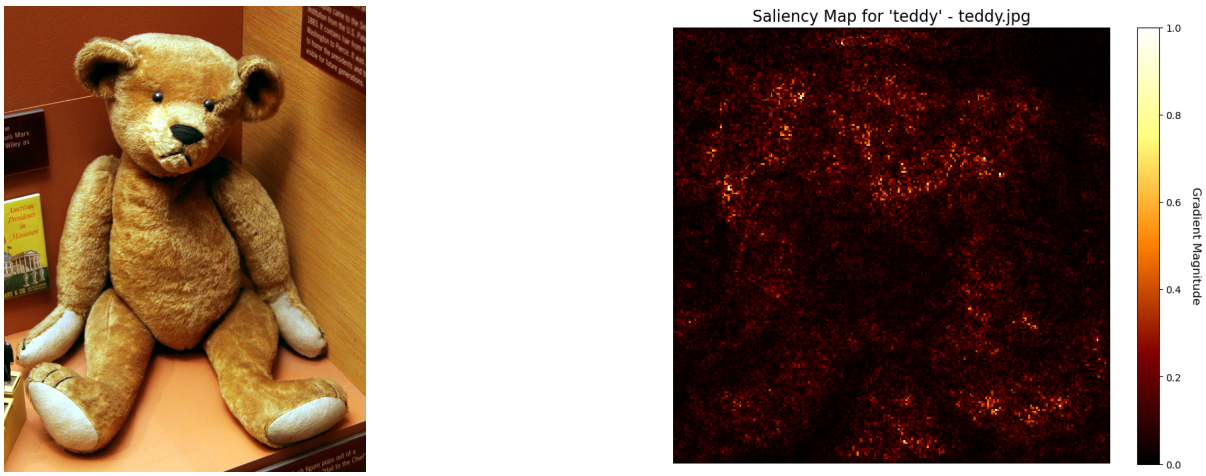

Figure 9: On the left: Original image of the teddy. On the right: Saliency map for the image.

References

[1] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*, ICLR Workshop 2014, https://arxiv.org/pdf/1312.6034.

[2] Russakovsky, O., Deng, J., Su, H., *ImageNet Large Scale Visual Recognition Challenge*, International Journal of Computer Vision (IJCV). Vol 115, Issue 3, 2015, pp. 211-252, https://arxiv.org/pdf/1409.0575.

[3] Simon J.D. Prince, *Understanding Deep Learning*, MIT Press, 2023, https://udlbook.github.io/udlbook/.

[4] Michael A. Nielsen, *Neural Networks and Deep Learning*, Determination Press, 2015, http://neuralnetworksanddeeplearning.com/.