TRANSFER LEARNING AND SALIENCY MAP VISUALIZATION

Assignment 4, CS-587

Author
Nikolaos Barmparousis
csd4690
May 5, 2024

Contents

1 Introduction

This assignment explores transfer learning and saliency map visualization. In Part A, we fine-tune a pre-trained AlexNet model on the WikiArt dataset to classify paintings by style. In Part B, we use a pre-trained VGG-16 model to generate image-class saliency maps, highlighting relevant image regions.

2 Part A: Transfer Learning

Transfer learning is a technique in machine learning where a pre-trained model is adapted to solve a new but related task. By leveraging the knowledge learned from a large source dataset, the model requires less data and time for training on the target dataset. In this assignment, we use a pre-trained AlexNet model, initially trained on ImageNet, to classify paintings in the WikiArt dataset, thereby reducing the computational cost and improving the efficiency of training.

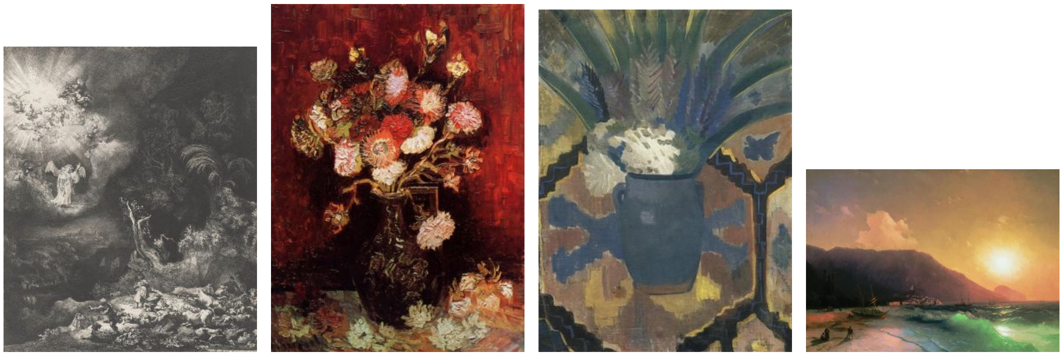
In the following sections, we describe the WikiArt dataset and the modifications made to the AlexNet model. We then discuss the fine-tuning setup, present the training results, and answer theoretical questions related to transfer learning.

2.1 Dataset and Pretrained Model

Dataset: WikiArt

The WikiArt dataset consists of 4000 paintings categorized into 10 artistic styles, including Baroque, Realism, Expressionism, and Romanticism. Each painting is resized to a standard input size suitable for the AlexNet model. The small size of the dataset may impose some training difficulties, especially if it doesn't share some structure with the ImageNet dataset.

Figure 1 provides sample paintings from four artistic styles in the WikiArt dataset.



Baroque

Realism

Expressionism

Romanticism

Figure 1: Sample paintings from the WikiArt dataset showcasing four artistic styles.

Pretrained Model: AlexNet

AlexNet is a Convolutional Neural Network (CNN) trained on the ImageNet dataset, containing over a million images across 1000 categories. The model comprises several convolutional layers for feature extraction, followed by a classifier section with fully connected (FC) layers.

Figure 2 shows an overview of the AlexNet architecture.

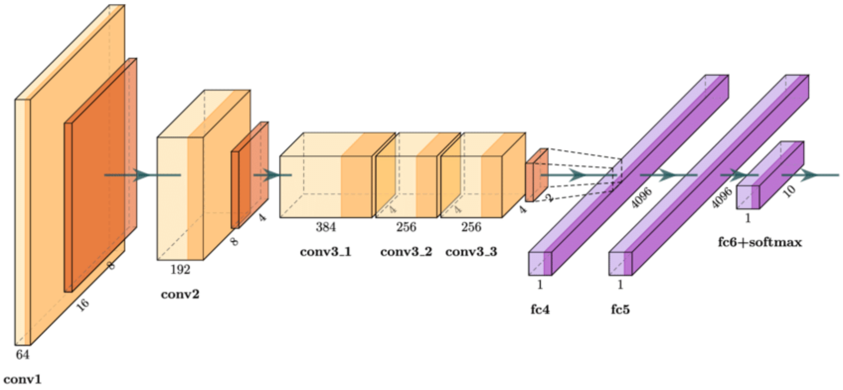


Figure 2: AlexNet architecture showing convolutional and fully connected layers.

2.2 Fine-Tuning Setup

In this assignment, we modify the classifier section of AlexNet to classify paintings based on artistic styles in the WikiArt dataset. Two modifications are explored:

- **Replacing the Last Fully Connected Layer**: The final FC layer in AlexNet is replaced to match the number of classes in the WikiArt dataset.
- **Replacing the Last Two Fully Connected Layers**: The last two FC layers are removed and replaced with a single new layer, again matching the number of classes as its output.

Figures 3 and 4 illustrate the modifications made to the AlexNet classifier by visualizing the computational graph of the classifier section.

For training, we use the cross-entropy loss function and fine-tune the pre-trained AlexNet model using stochastic gradient descent (SGD) with a learning rate of 0.01 and momentum of 0.9. The model is trained for 5 epochs, monitoring training and validation accuracy throughout.

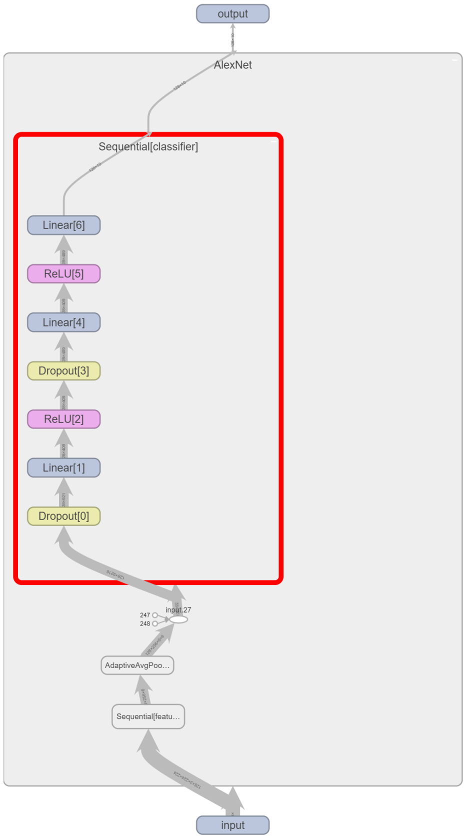


Figure 3: AlexNet Classifier with Last Fully Connected Layer Replaced

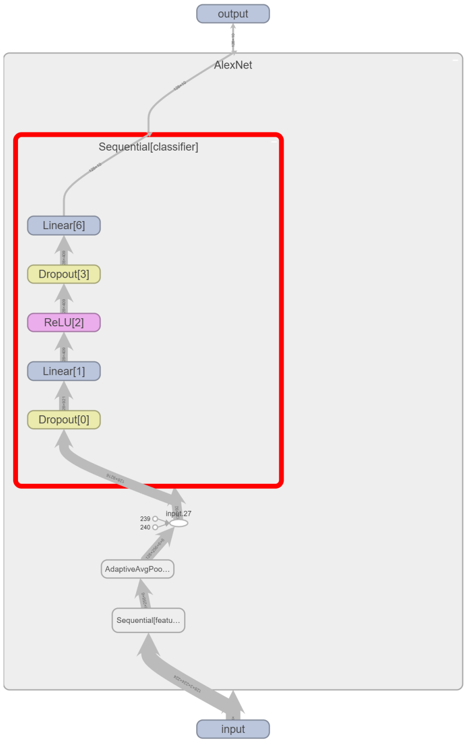


Figure 4: AlexNet Classifier with Last Two Fully Connected Layers Replaced

2.3 Results

We analyzed the effectiveness of two fine-tuning strategies on the WikiArt dataset, observing differences in training loss and validation accuracy. Figure 5 shows the validation accuracy for each epoch, while Figure 6 displays the training loss. The strategy involving the replacement of the last two fully connected layers reached a peak accuracy of 0.386 at epoch 4, compared to 0.368 at epoch 5 for the strategy that only replaced the last layer.

The modest accuracy figures highlight the challenges of applying transfer learning when the source and target datasets are fundamentally different. ImageNet features real-world images across 1000 classes, which contrasts sharply with WikiArt's abstract artistic styles. Although ImageNet provides a rich feature set, these features translate poorly to the abstract and varied styles of WikiArt, limiting the effectiveness of the transferred learning.

2.4 Discussion on Transfer Learning

Given the noticeable differences between the ImageNet and WikiArt datasets and our model's subpar performance, it's important to consider different scenarios where the nature and size of the dataset can influence the decision to fine-tune or train from scratch.

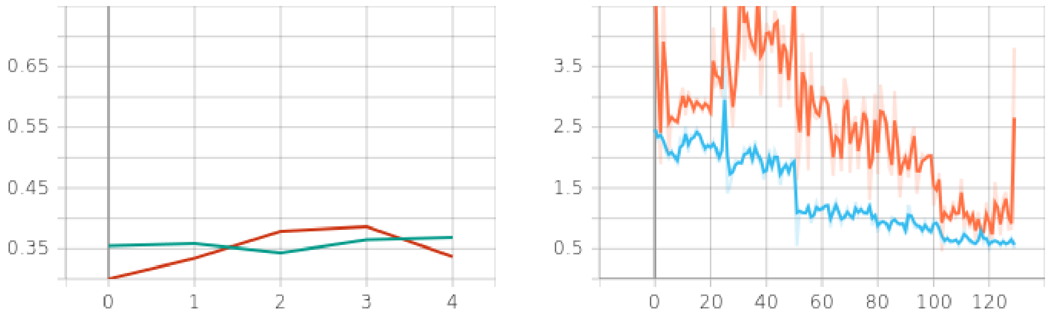


Figure 5: Validation accuracy per epoch

Figure 6: Training loss per epoch

Figure 7: Validation Accuracies and Training Loss plots for the 2 configurations. 1 layer replaced is blue, 2 layers replaced is red.

- **My dataset is small but similar to the original dataset. Should I fine-tune?**
  Fine-tuning is both necessary and appropriate here, as the small size of the target dataset benefits from the pre-established network weights that already understand similar features.

- **My dataset is large and similar to the original dataset. Should I fine-tune or train from scratch?**
  Both strategies could be viable, and experimenting with each may be the best way to determine the optimal approach. Fine-tuning can capitalize on the existing relevant features for faster convergence, whereas training from scratch offers flexibility at the cost of increased complexity and resource consumption.

- **My dataset is different from the original. Should I fine-tune?**
  As seen with the WikiArt dataset, fine-tuning a model trained on a significantly different dataset like ImageNet may not yield the best results. The model's learned features from ImageNet were less applicable to the abstract styles of WikiArt, limiting the effectiveness of transfer learning.

3 Part B: Saliency Map Visualization

Saliency maps, as introduced in the work by Simonyan et al. (2014), help visualize which parts of an image influence a convolutional neural network's classification decision. Utilizing a pretrained VGG-16 network, we will visualize saliency maps for images of a husky, flamingo, doberman, teddy bear, and cat. The process involves:

1. Performing a forward pass of the image through the network.

2. Calculating the scores for every class.

3. Enforcing the derivative w.r.t. the input image of the score vector at the last layer to be zero for all classes except for the class of interest, which is set to one.

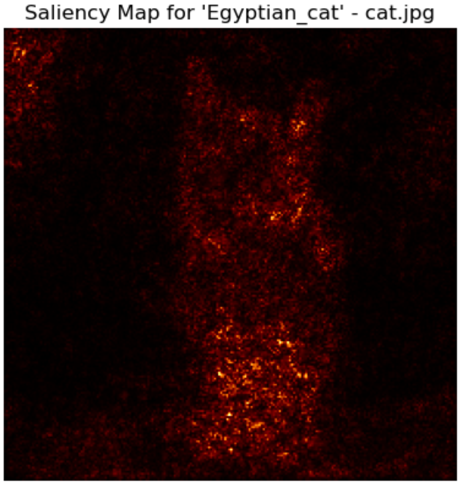
4. Back-propagating this derivative to the input image level.

5. Rendering the gradients to obtain the Saliency Map.

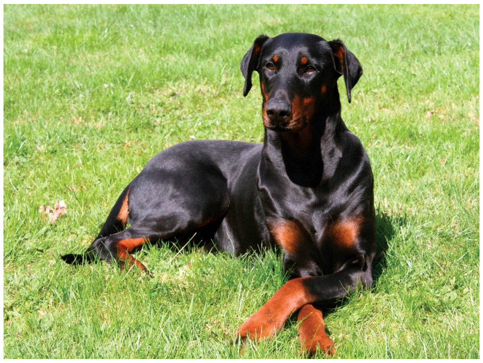
This section will present the results from this procedure, highlighting how the model focuses on different aspects of each image.

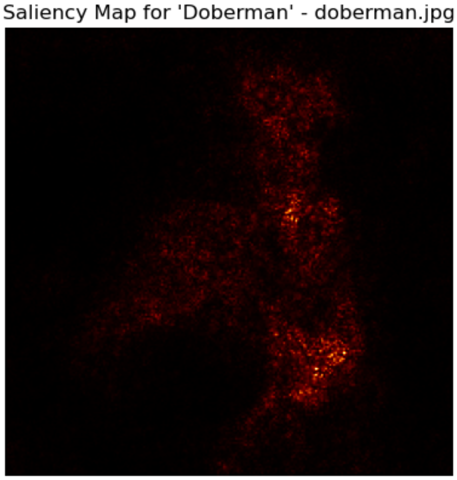
3.1 Saliency Map Visualizations and Analysis

The model correctly focuses on the body of the cat (feet, ears, and body) but misclassifies the image as an Egyptian cat, whereas we have a simple household cat as the image.



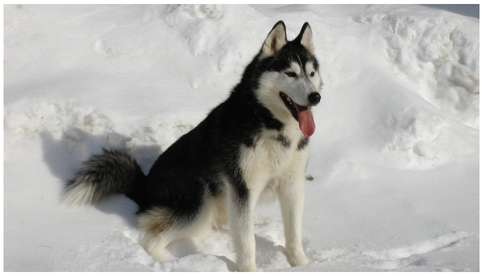
For the Doberman and the Husky, the model correctly focuses on the body of the animals, with a specific emphasis on the legs of the Doberman and the hairy coat of the Husky.

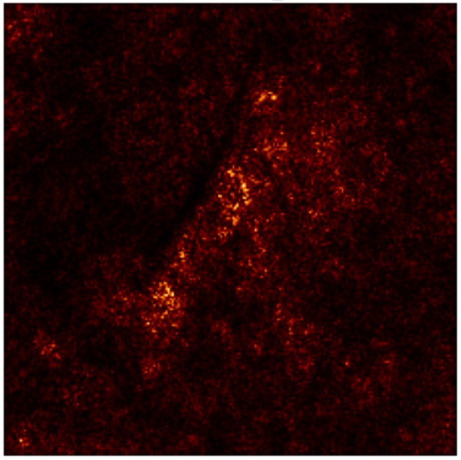




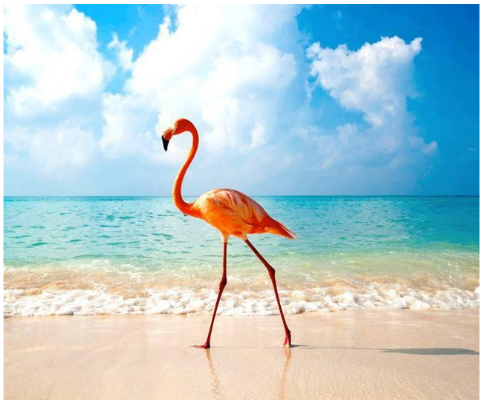
While the model's main focus is on the Flamingo, it incorrectly classifies it as a seashore.

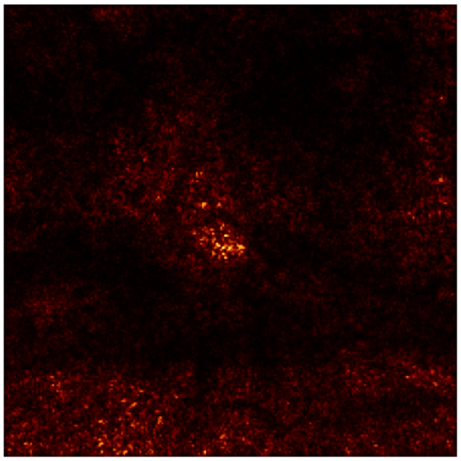
For the Teddy Bear, the model primarily focuses on its head, correctly classifying it.



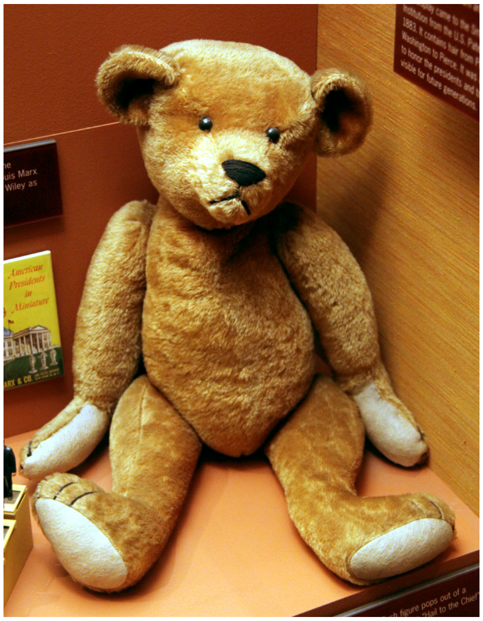


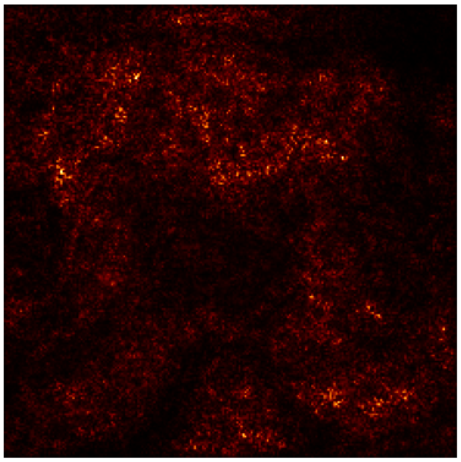
Saliency Map for 'Eskimo_dog' - husky.jpg





Saliency Map for 'seashore' - flamingo.jpg





Saliency Map for 'teddy' - teddy.jpg