Advanced Topics in Machine Learning

Assignment 0 - Report

8 September 2025

1 Introduction

This assignment includes key concepts in the course, beginning with CNN architectures that use skip connections, such as ResNet-152, and progressing toward more advanced models, including Vision Transformers (ViT), Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and the multimodal CLIP model (Contrastive Language-Image Pretraining). The first task includes fine-tuning a model's task head and comparing the use of pretrained weights versus randomly initialized ones while also building an understanding of transfer learning. Task 2 involves understanding the internals of the Vision Transformer, involving visualization of attention maps and experimentation with both random and structured masking strategies. Lastly, in task 4, the goal is to train a VAE on the MNIST dataset, with the main outcomes being reconstruction, generation, and an analysis of posterior collapse.

1.1 Task 1: Inner Workings of ResNet-152

ResNet-152 is a deep convolutional neural network that is based on an idea called skip connections or residual connections. Usually, when a network becomes very deep, it faces problems like vanishing gradients, which makes training difficult. ResNet-152 solves this by adding input directly to the output of a layer with skipping a few layers. In this way, when the back propagation occurs, the gradients does not face vanishing. With 152 layers, it is one of the deeper versions of ResNet and is used or tasks like image classification, object detection, and feature extraction.

1.1.1 Methodology

First, the ResNet-152 model was taken from PyTorch with default pretrained weights. The CIFAR-10 dataset was used, but since its images are 32×32 and ResNet-152 is trained on ImageNet with 224×224 images, the CIFAR-10 images were resized to 224×224. The classification head was then trained on this dataset while keeping the pretrained weights, but the results after 10 epochs were not very good. Later, the backbone of the model was frozen, and only the classification head was fine-tuned. In the next step, the skip connections in the Bottleneck class were removed by making a new class without them and replacing some layers to test the change. After that, features from early, middle, and late layers were collected by running the model for one epoch and then visualized using t-SNE. Finally, experiments were done on transfer learning by fine-tuning the model with both pretrained and random weights and also by comparing full fine-tuning with only classification head fine-tuning.

1.1.2 Results

Here is a comparison of the results: In the baseline setup, the classification head was fine-tuned on CIFAR-10. With the original 32×32 image size, the results were poor, but after resizing to 224×224, the performance improved because the larger size provided a richer feature space. When residual connections were removed, the model performed worse at first but showed slow improvement over 5 epochs. For feature hierarchies, early, middle, and late features were extracted, and the late ones were plotted to visualize how the model learns representations. In transfer learning, both full fine-tuning and classification-head-only fine-tuning were tested with default and random weights. In both cases, using default pretrained weights gave much better results, showing that pretraining helps the model generalize well to new datasets.

- **Without Upsampling:** *Epoch 10: train_loss=1.7791 train_acc=0.3870 — val_loss=1.7156 val_acc=0.4307*
- **With Upsampling:** *Epoch 10: train_loss=0.3677 train_acc=0.8764 — val_loss=0.3943 val_acc=0.8717*

1.1.3 Discussion

The results show that resizing CIFAR-10 images to 224×224 improved performance, showing its importance for input dimensions that match the pretrained model. Disabling residual connections reduced accuracy, highlighting their importance in improving deep network training. Transfer learning with pretrained weights outperformed random initialization, showing the benefit of reusing knowledge from large datasets. However, the study is limited by the small dataset, few training epochs, and partial removal of skip connections, so broader experiments are needed for stronger conclusions.

1.2 Task 2: Understanding Vision Transformers (ViT)

1.2.1 Methodology

I selected the DeiT-Small model, pre-trained on ImageNet-1k from HuggingFace. It takes input images in the shape 224x224. After loading the model, I downloaded sample image of dog and car from the internet and took inference. The prediction seems reasonable as both the predicted labels matches the image. I than extracted the attention weights for the test image. Then aggregated across heads with mean and got one clean attention map. As the input image is

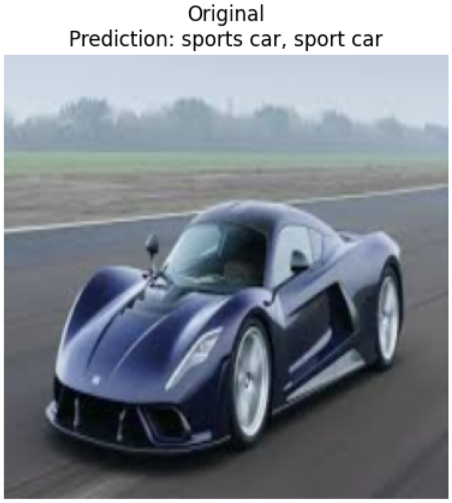


Figure 1: Original Image

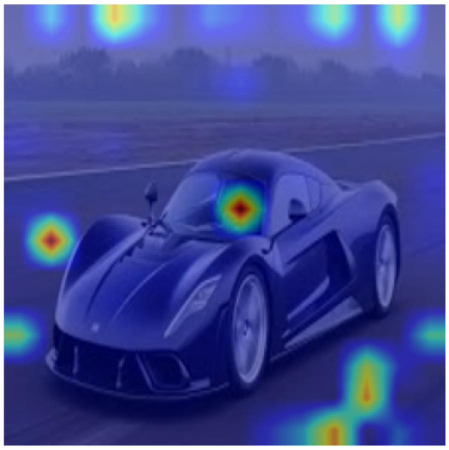


Figure 2: Attention on Car

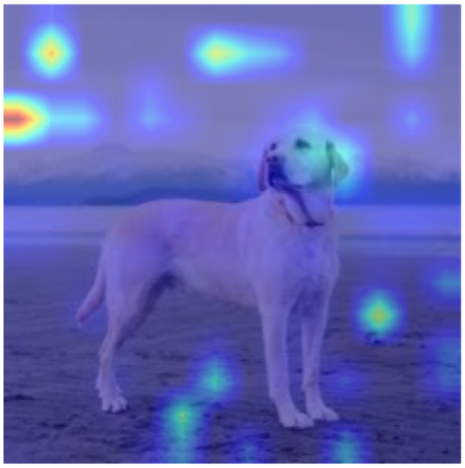


Figure 3: Overlayed Attention Image

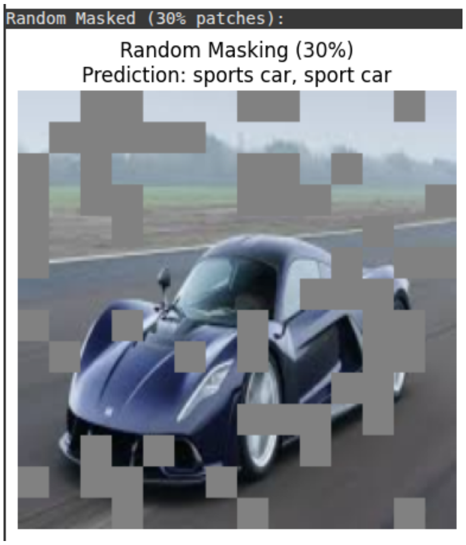
224×224 pixels and each patch is 16×16, the image is divided into 14 patches. So, the whole image becomes a grid of $14 \times 14 = 196$ patches. When we take the CLS token's attention scores, we get a vector of 196 values which is one score per patch. If we reshape this vector back into a 14×14 grid, we recover the original spatial layout of patches in the image. Than upsampled the attention map to image resolution and overlayed it on the input image.

As seen from the image, the attention focueses on the key parts to identify the shape of the car. It did focussed on other parts of the image as well due to preprocessing issues in the input image. But most of the attention focuses on the key parts of car's shape. The model now identified the car as sports car, still correctly able to classify.

1.3 Results and Discussion

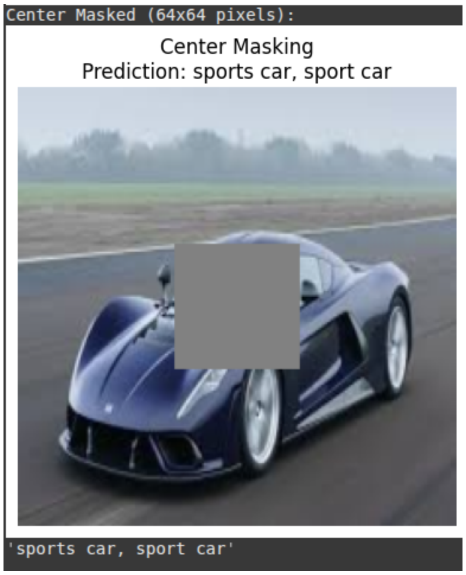
On applying center masking the model identified the image as a Sports Car, which is again correct. By applying masking the model able to guess the label correctly which means that the model is robust to missing patches. ViTs use global self-attention and distribute attention across many patches, allowing them to correctly classify even when important regions are masked.

In my case, both the CLS token and mean pooling gave the same accuracy, 95 percent, which means the model has learned strong and consistent global representations. The CLS token is designed to gather global information during pretraining, while mean pooling takes an average over all patch embeddings to capture the overall image equally. Since the model performs equally well with both, it shows that the pretrained ViT distributes information across both the CLS token and the patch tokens effectively.



```
Random Masked (30% patches):
```

Figure 4: 30 percent Masking



```
Center Masked (64x64 pixels):
```

```
'sports car, sport car'
```

Figure 5: Center Masking

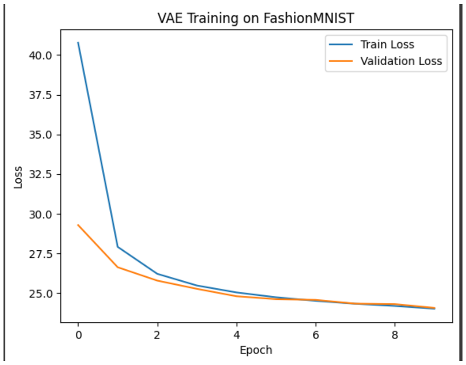


Figure 6: Training VAE

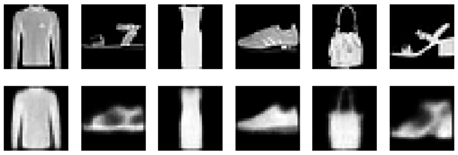


Figure 7: Visualize Reconstruction

1.4 Task 4: Training Variational Autoencoders

1.4.1 Methodology

I trained a Variational Autoencoder (VAE) on the FashionMNIST dataset using the provided architecture. The dataset was loaded from torchvision datasets, and the model structure was imported from the architecture.py file without any modifications. The training objective combined two components: the mean squared error (MSE) loss for image reconstruction quality and the KL divergence. This composite loss corresponds to the standard Evidence Lower Bound (ELBO) objective for VAEs. The model was trained for 20 epochs, and both training and validation losses showed a consistent downward trend. By the final epoch, the VAE achieved a training loss of 23.1087 and a validation loss of 23.3124, indicating stable learning and good generalization between training and validation sets.

To evaluate the performance of the trained VAE, I visualized reconstructions of FashionMNIST test images by passing them through the encoder–decoder pipeline. The encoder maps the input image to a latent distribution, from which a latent sample is drawn using the reparameterization trick. The decoder then reconstructs the image from this latent representation. As shown in the figure, the reconstructed images (bottom row) preserve the overall structure and outline of the original test samples (top row), capturing key features such as clothing shape and texture. Although some fine details are blurred, and the black spots are removed.

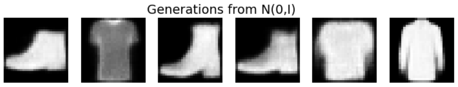


Figure 8: Visalizing Generations

To assess the generative ability of the VAE, I sampled latent vectors p(z)=N(0,I) and passed them through the decoder to produce new images. The generated samples resemble realistic FashionMNIST items such as shirts, shoes, and bags, demonstrating that the VAE learned to map random points in the latent space to meaningful image structures.

1.5 Results and Discussion

I then generated new samples by drawing latent vectors z from a Laplacian distribution. Compared to the Gaussian, the Laplacian has heavier tails, which means it samples more extreme latent values. As a result, the generated images show greater variation in style and intensity, though some samples appear noisier or less coherent. This highlights how the choice of prior distribution influences the diversity and sharpness of generations, with Gaussian priors producing smoother results and Laplacian priors introducing more variability.

2 Conclusion

In this report, I explored different deep learning models including ResNet, Vision Transformers, and VAEs. The experiments showed the importance of skip connections, pretrained weights, and input size for ResNet. For Vision Transformers, attention maps and masking experiments proved their robustness to missing patches. The VAE was able to reconstruct and generate images, and the choice of prior affected the diversity of outputs. Overall, these tasks gave practical understanding of how modern models work and where they can be improved.

GitHub Repository