Artificial Neural Networks Project: Scene Classification

Niels Van den Broeck

May 30, 2025

1 Introduction

In this report, I will go over the design choices and results of the artificial neural networks project: Scene Classification. We used the given **15-Scene dataset**, a benchmark image classification dataset that consists of 15 distinct scene categories, including both indoor and outdoor environments such as *office*, *bedroom*, *mountain*, and *highway*. Each category contains a few hundred colored images, with different lighting conditions and viewpoints. The diversity of the dataset makes it well-suited for evaluating representation learning methods.

We explore and compare three different learning paradigms for image classification:

- **Vanilla Supervised Learning:** A standard convolutional neural network (ResNet-18) trained end-to-end using cross-entropy loss with class labels.
- **SimCLR (Simple Contrastive Learning of Representations):** An unsupervised contrastive learning framework where the model is trained to bring augmented views of the same image closer in the feature space, followed by a linear probe trained on frozen encoder features.
- **SupCon (Supervised Contrastive Learning):** A supervised contrastive approach that extends SimCLR by using label information during contrastive pretraining to pull together all examples of the same class, enhancing class-specific clustering in the feature space.

The goal of this project is to evaluate the effectiveness of these methods on the 15-Scene dataset and analyze their performance using classification accuracy and visualizations —such as t-SNE and confusion matrices— in section 5.

2 Supervised Learning

In the supervised learning setup, the model is trained end-to-end using labeled image data. Each input image is associated with a ground-truth class label, and the model learns to minimize the cross-entropy loss between its predictions and the true labels.

We implement a vanilla supervised learning baseline using a ResNet-18 architecture. The model is initialized from scratch (i.e., without pretrained weights [1]), and trained directly on the 15-Scene dataset using the Adam optimizer. The final classification layer consists of a fully connected layer with 15 output units, referring to the different classes. During training, we monitor validation accuracy and apply model checkpointing to keep the best-performing model.

3 Contrastive Pretraining

Contrastive learning is a self-supervised approach where the model learns to distinguish between similar and dissimilar image pairs in the absence of explicit labels. The goal is to learn a representation space where semantically similar images are closer together and dissimilar ones are farther apart. This is typically achieved by generating multiple augmented views of the same image and training the model

[1] This is done in every experiment to have consistent comparisons between the models.

to minimize the distance between representations of the same image while maximizing the distance to others in the batch.

In this project, we use two contrastive learning approaches: SimCLR and SupCon. Both methods rely on a ResNet-18 encoder followed by a projection head to map features into a contrastive embedding space. After pretraining, a linear classifier is trained on top of the frozen encoder for scene classification in section 4.

3.1 SimCLR

SimCLR (Simple Contrastive Learning of Representations) is a self-supervised method that generates positive pairs by applying two random augmentations to each image. During training, the model learns to associate these augmented views while treating all other images in the batch as negatives. The model is trained using the NT-Xent (Normalized Temperature-scaled Cross Entropy) loss function.

3.2 SupCon

SupCon (Supervised Contrastive Learning) extends SimCLR by leveraging label information during contrastive pretraining. Instead of treating only augmented views of the same image as positives, SupCon considers all samples in the batch that belong to the same class as positive pairs. This leads to more semantically meaningful clusters in the feature space. Our implementation follows the same architectural setup as SimCLR, using ResNet-18 and a projection head. During training, we apply the supervised contrastive loss.

4 Linear Probe

After contrastive pretraining, we evaluate the quality of the learned feature representations by training a linear classifier on top of the frozen encoder. This approach allows us to assess how well the encoder has organized the feature space without further fine-tuning the backbone network.

In our implementation, we freeze the weights of the ResNet-18 encoder obtained from either SimCLR or SupCon pretraining. We then train a single fully connected (linear) layer that maps the 512-dimensional encoder output to the 15 scene classes. The classifier is trained using the cross-entropy loss and the Adam optimizer. We monitor validation accuracy throughout training and save the best-performing model checkpoint. This setup ensures a fair comparison between representations learned via SimCLR and SupCon, as the evaluation focuses solely on the linear separability of the encoded features.

5 Results

In table 1, the different settings with corresponding hyperparameters are compared. Due to hardware limitations, all models were trained for 50 epochs. This means, for SimCLR and SupCon, 50 epochs are trained plus an extra 50 epochs for the Linear Probe. The batch size was set to the maximum that could fit in GPU memory: 64 for both models. We used the Adam optimizer due to its effectiveness in deep learning tasks. Finally, a learning rate of 1×10^{-3} is used, which is a common default setting for training deep networks.

Model	**Supervised**	**SimCLR + Linear Probe**	**SupCon + Linear Probe**
Epochs Trained	50	50 + 50	50 + 50
Batch Size	64	64	64
Learning Rate	1e-3	1e-3	1e-3
Optimizer	Adam	Adam	Adam
- Test Accuracy	87.41%	76.57%	80.67%

Table 1: Training configurations and test accuracy for the three evaluated models. Contrastive methods use a frozen encoder with a linear classifier trained post-hoc.

5.1 Supervised

As you can see in figure 1, training with the vanilla supervised technique results in high performance, reaching an astonishing validation accuracy of 87.41%. In the plots, it stands out that the validation curves are not stable at all and include a lot of spikes, In contrast, the validation curves for SimCLR and SupCon were much smoother (see later). This difference can be attributed to the nature of training in each setting. In the supervised case, the model is updated end-to-end using label-supervised gradient signals. As a result, any noisy gradient updates or overfitting to mini-batch patterns can directly affect the encoder's feature space and the classifier simultaneously. This leads to higher sensitivity in validation performance from one epoch to the next, particularly when training for a limited number of epochs without regularization or learning rate scheduling.

Looking at the t-SNE visualization in figure 2, a clear distinction is made between the different classes, as well as each class being located in a compact manner. These are desired results and confirm the high accuracy.

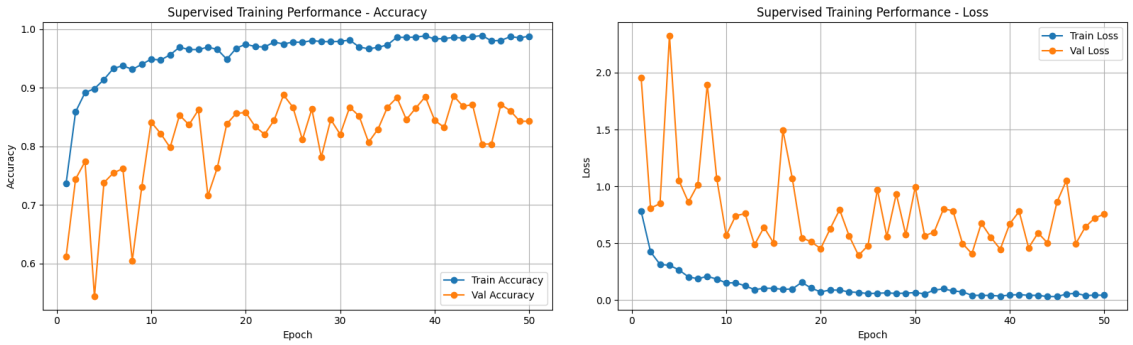

Figure 1: Training and validation accuracy (left) and loss (right) for the Supervised model.

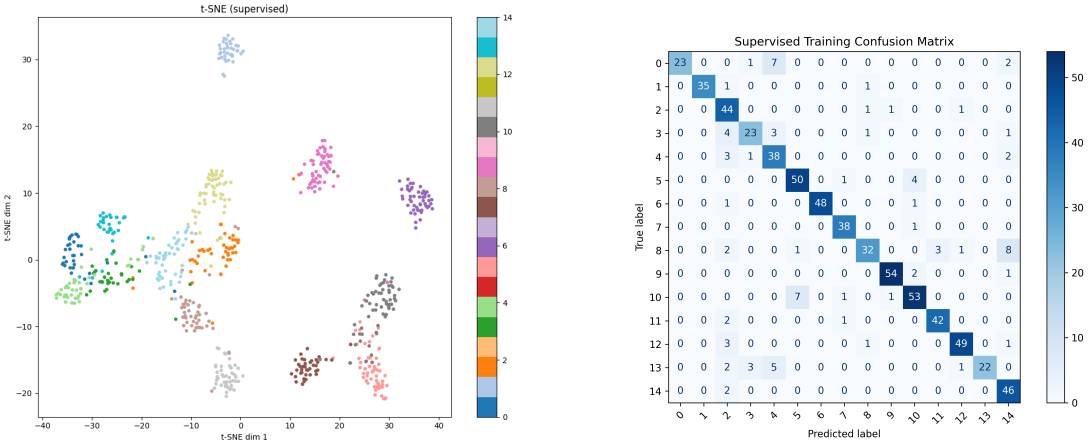

Figure 2: t-SNE visualization (left) and confusion matrix (right) for the Supervised model.

5.2 SimCLR + Linear Probe

Among the three approaches evaluated, the SimCLR model achieved the lowest classification accuracy. This outcome is expected due to its limitations.

Most importantly, SimCLR is trained without any label supervision. It relies entirely on data augmentations to generate positive pairs and distinguishes between samples using only their visual appearance. While this approach enables the model to learn general-purpose features, it does not guarantee that these features will be linearly separable with respect to the downstream classification task.

In addition, the effectiveness of SimCLR is highly dependent on the batch size, which controls the number of negative samples used in the contrastive loss. Due to hardware constraints, we were limited to a batch size of 64. This is small for contrastive learning, where batch sizes of 256 or larger are commonly used to ensure rich and diverse negative pairs. A smaller batch size weakens the contrastive signal, making representation learning less effective.

Despite these limitations, SimCLR still produced a reasonable test accuracy and provides a useful baseline for unsupervised learning. Its performance confirms the strength of contrastive methods even in the absence of labels, while also highlighting the value of supervision in tasks like scene classification.

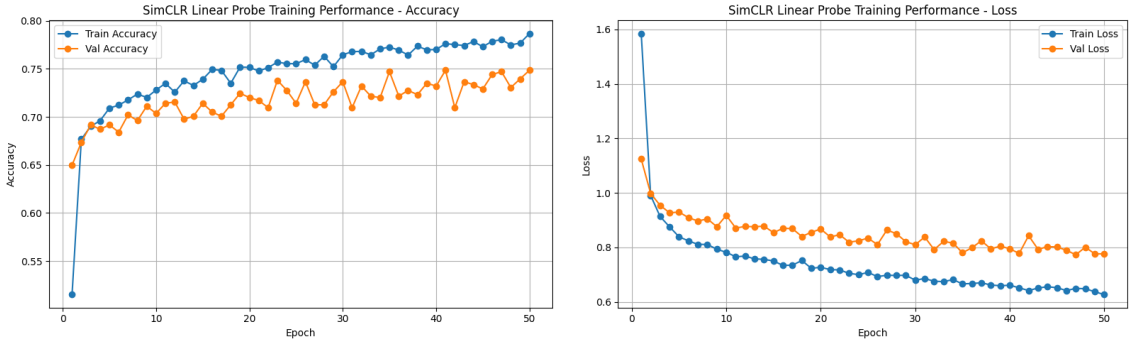

Figure 3: Training and validation accuracy (left) and loss (right) for the SimCLR + Linear Probe model.

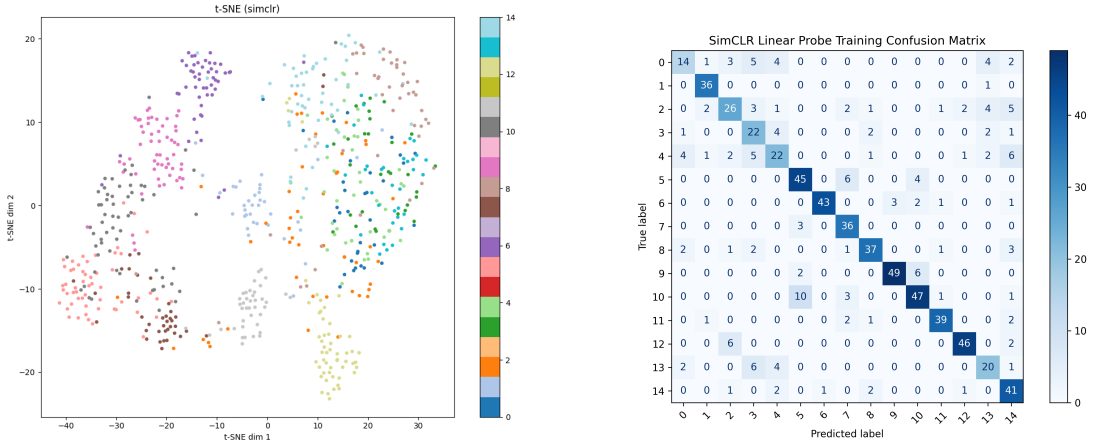

Figure 4: t-SNE visualization (left) and confusion matrix (right) for the SimCLR + Linear Probe model.

5.3 SupCon + Linear Probe

Compared to SimCLR, the SupCon method achieved a slightly higher accuracy in our experiments. This improvement is consistent with the design and objectives of supervised contrastive learning.

SupCon leverages label information during pretraining. Specifically, SupCon pulls together not only augmented views of the same image but also all other images in the batch that share the same class label. This creates stronger and more semantically meaningful feature clusters, where samples from the same class are grouped more tightly in the representation space. As a result, the representations learned by SupCon are inherently more aligned with the downstream classification task. This is particularly advantageous when evaluating with a linear probe, as the improved class separability in the embedding space makes the linear decision boundary more effective.

In figure 6, you can see that the different classes are mostly separated. However, in some cases, the classes still overlap. For example, when looking at the t-SNE plot, in the top left there are a lot of

pink (label 5) and gray (label 10) points colliding. This is also clearly visible in the confusion matrix where 11 samples were predicted as class 5, while actually being in class 10. This is simply the result of too similar classes that that the model cannot distinguish between yet.

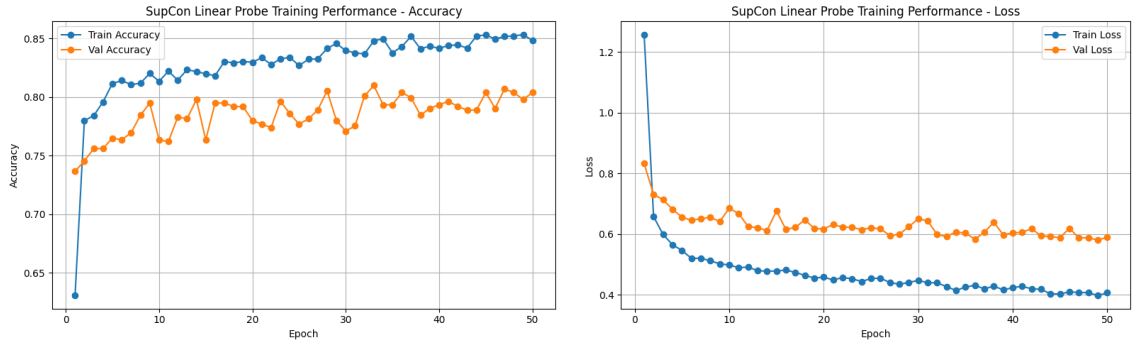

Figure 5: Training and validation accuracy (left) and loss (right) for the SupCon + Linear Probe model.

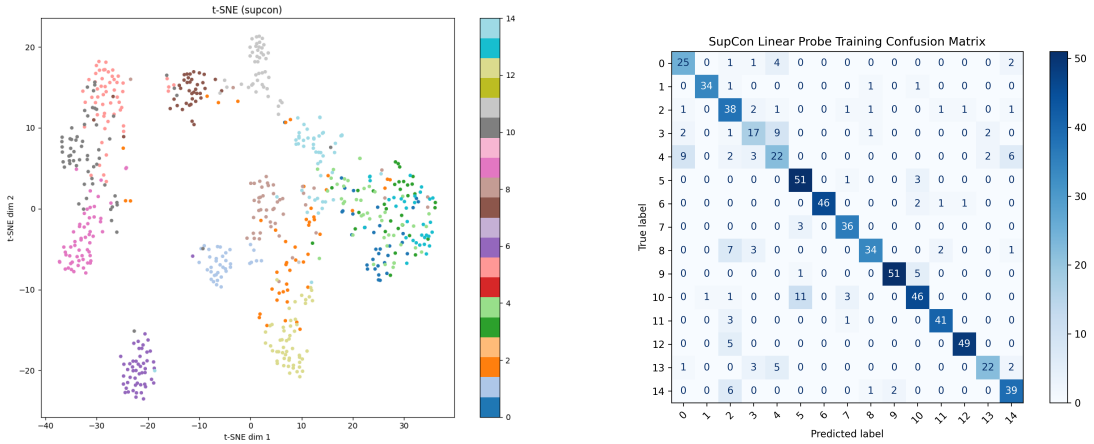

Figure 6: t-SNE visualization (left) and confusion matrix (right) for the SupCon + Linear Probe model.

5.4 Managing under- and overfitting

To monitor and manage potential underfitting or overfitting, we evaluated each model using validation accuracy and loss curves across training epochs. The supervised model, trained end-to-end, showed relatively stable convergence without significant signs of overfitting. This was evident from the fact that the validation accuracy closely followed the training accuracy throughout training, and the validation loss decreased steadily alongside training loss (Figure 1).

For SimCLR and SupCon, we trained a linear probe on top of a frozen encoder. Since only the linear layer was updated during this phase, the risk of overfitting was inherently low. This is supported by the smooth and consistent validation curves, with no significant divergence between training and validation performance (Figures 3 and 5).

We mitigated overfitting by using strong data augmentation during training (especially for contrastive learning), and by limiting training to a moderate number of epochs. The batch size was kept as large as the hardware allowed, which also helped stabilize training and generalization. Overall, the models showed no substantial signs of underfitting or overfitting under these settings.