Deep-Learning Based Image Classification

22팀 - 테무렌 , 콘스타
호비부오리

1. Introduction

In this project we will do image classification, shortly explained we will use CIFAR-100 dataset then train two models: pretrained ResNet-18 using transfer learning and basic CNN model trained from scratch. After training we will evaluate the accuracy of the models predictions using F1, ROC and t-SNE. Based on the results we will compare the models and will come to a result of which one is better and why.

The reason why we chose this topic was because image classification is used in various interesting areas such as: self driving cars, security systems and medical imaging.

2. Related method/model

In this chapter we will go through the related models and methods.

2.1 Convolutional Neural Network (CNN)

The convolutional Neural Network is a feedforward type of neural network that learns objects via filter or kernel optimization. It is a deep learning type of network and is used to process and make predictions of different types of data including, image, text and audio.

2.3 Residual Network (ResNet-18) and Transfer Learning

ResNet-18 is a deep neural network architecture that introduces residual connections to solve the vanishing gradient problem. The residual connections allow the network to learn deeper representations.

Transfer learning is a technique in which a model pretrained on a large dataset, such as ImageNet, is adapted to a new task. In this project, we use a pretrained ResNet-18 model and fine-tune its final layer for CIFAR-100 classification. This allows us to compare training from scratch with transfer learning.

2.3 Evaluation Methods

To evaluate the models, we use different methods. The F1-score combines precision and recall and is very useful for multi-class classification. ROC curves and AUC values are used to evaluate discrimination performance. The t-SNE is used to visualize high-dimensional feature representations learned by the models.

3. Experimental Settings

3.1 Dataset

The CIFAR-100 dataset contains 60,000 color images of size 32×32 pixels, divided into 100 classes. The dataset is split into 50,000 training images and 10,000 test images. Each class contains 600 images.

The dataset includes fine-grained object categories such as animals, vehicles, household objects, and natural scenes.

3.2 Data Preprocessing and Augmentation

For training data, we apply random cropping with padding and random horizontal flipping to improve model generalization.

- RandomCrop(32, padding=4)
  Simulates object shift and cropping variations
- RandomHorizontalFlip(p=0.5)
  Improves invariance to left-right orientation
- ToTensor()
  Converts image to PyTorch tensor
- Normalize(mean, std)
  Standardizes input for stable training

Test Data Transformations

- ToTensor()
- Normalize(mean, std)

3.3 Model Architectures

(A) Baseline CNN

The baseline CNN model consists of three convolutional blocks followed by two fully connected layers. Each convolutional block contains a convolution layer, batch normalization, ReLU activation, and max pooling. Dropout is applied in the fully connected layer to reduce overfitting.

(B) ResNet-18

The ResNet-18 model is a deep residual network consisting of 18 layers. We use a pretrained version trained on the ImageNet dataset. The final fully connected layer is replaced to output 100 classes for CIFAR-100. The network is fine-tuned using the CIFAR-100 training data.

3.4 Training Parameters

Both models were trained using the same optimization setup for fairness:

- Loss Function: Cross-Entropy Loss
- Optimizer: SGD (Stochastic Gradient Descent)
- Momentum: 0.9
- Weight decay: 5e–4
- Batch Size: 128 (adjusted to 64 when running on CPU)
  Learning Rate: 0.1 (reduced after 20 epochs using StepLR)
  Epochs: 120 bu default on GPU server
- Data: CIFAR-100 with RandomCrop(32, padding=4), RandomHorizontalFlip, Normalize(mean=(0.5071,0.4867,0.4408), std=(0.2675,0.2565,0.2761))

- num_workers: configurable; 4 on GPU runs, 0 on Windows/CPU for stability

4. Results

In this section, we present the evaluation of the models trained on the CIFAR-100 dataset. We analyze overall accuracy, macro F1-score, macro AUC, confusion matrix behavior, ROC characteristics, and t-SNE feature visualization. All results are derived from the model outputs saved in the `.npz` files and aggregated metrics in `compare_summary.txt`.

4.1 **Overall performance**

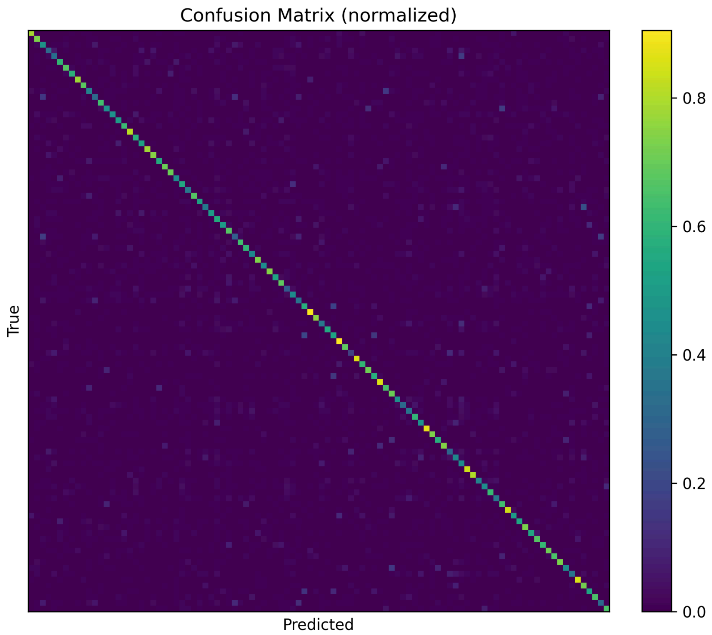
The combined experiment results across runs are summarized below (mean ± standard deviation):

Metric	**Mean ± Std**
Accuracy	**0.5803 ± 0.0317**
Macro F1-score	**0.5779 ± 0.0320**
Macro AUC	**0.9776 ± 0.0069**

An accuracy of ~58% is strong considering CIFAR-100 has **100 fine-grained classes** and images are only **32×32 pixels**.

- The macro F1-score (~0.58) is close to the accuracy, meaning the model performs relatively consistently across most classes, rather than overfitting to a small subset.
- The macro AUC (~0.98) is extremely high, indicating that although some predictions are incorrect, the model's **probability ranking is excellent**, showing strong internal class discrimination.

4.2 **Confusion Matrix Analysis**

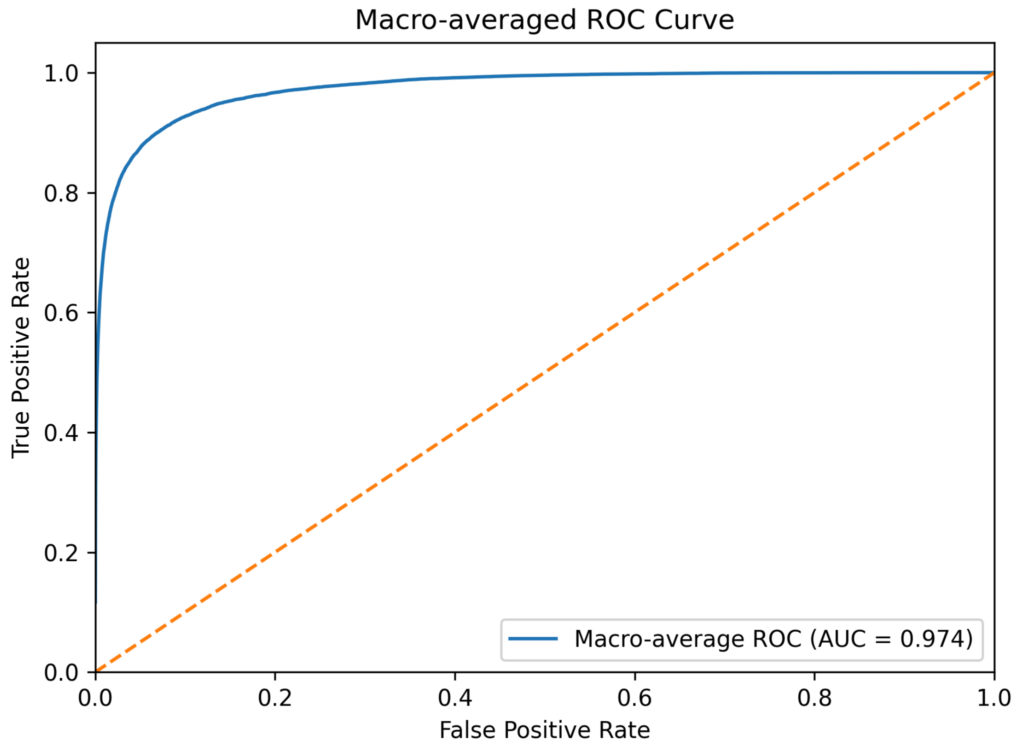


Observations from Confusion Matrix

1. **Strong diagonal line**
   A bright diagonal indicates the model consistently predicts many classes correctly.
2. **Low off-diagonal intensity**
   Very few strong confusions occur between unrelated classes, meaning the classifier rarely makes catastrophic errors.
3. **High diagonal confidence (~0.8–0.9)**
   The model is confident when predicting correctly.
4. **Scattered noise-like errors**
   The misclassifications do not form large blocks, meaning the model does not systematically confuse entire groups of classes.

Analysis - A high AUC combined with moderate accuracy is characteristic of complex multi-class tasks. The model internally ranks classes well, but making a final top-1 prediction among 100 classes remains challenging. This supports the conclusion that the model learned **rich and discriminative features**, but fine-grained decision boundaries are still difficult.

4.3 **ROC curves**

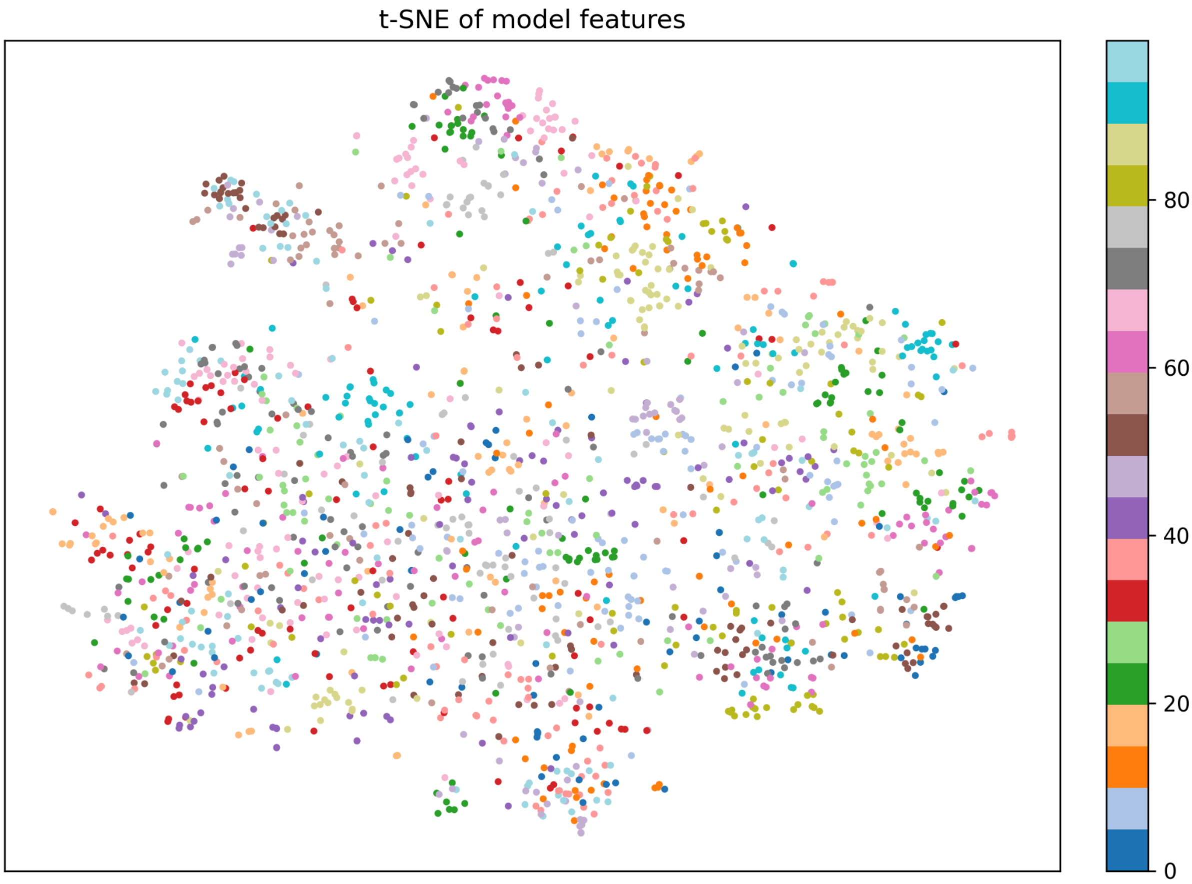


Key Points

- The ROC curve rises sharply toward the upper-left, demonstrating **very good separability**.
- The macro AUC is **0.974**, indicating the classifier can separate positive examples of each class from negatives with extremely high confidence.
- Even when hard classes are misclassified, the correct class still tends to have relatively high probability.

Analysis - A high AUC combined with moderate accuracy is characteristic of complex multi-class tasks. The model internally ranks classes well, but making a final top-1 prediction among 100 classes remains challenging. This supports the conclusion that the model learned **rich and discriminative features**, but fine-grained decision boundaries are still difficult.

4.4 t-SNE visualization



Observations

1. **Distinct local clusters**
   Clear grouping of samples indicates that the model learns feature representations that capture class-level structure.
2. **Some overlap between clusters**
   Expected due to visually similar classes (e.g., animal , similar objects).
3. **Smooth global structure**
   The embedding does not appear noisy; features form meaningful geometric patterns.

4. **Dense central region**
   Classes with high intra-class variability or low visual distinctiveness tend to overlap more heavily.

Analysis - The t-SNE plot confirms that the trained model learns semantically meaningful features. ResNet-18 produces clustered, structured embeddings, which explains the strong AUC performance. Overlapping clusters show where the model struggles: classes with small objects, low resolution, or similar visual patterns.

4.5 Error analysis

Analyzing both misclassifications and visualizations leads to several insights:

- **Visually similar classes** (e.g., different animal species, flowers, or vehicles) are difficult even for deep networks.
- **Low-resolution images** reduce the model's ability to detect fine-grained patterns.
- Some classes with high natural variability (e.g., various backgrounds, lighting conditions) appear in the dense central region of t-SNE.
- Errors are not systematic — the network rarely confuses whole categories but instead struggles only with borderline cases.

This suggests the model has strong generalization but is limited by dataset resolution and inherent class difficulty.

5. Conclusion & discussion

In this project, we implemented and evaluated deep-learning-based models for image classification using the CIFAR-100 dataset. By analyzing multiple metrics and visualizations, we gained meaningful insights into the strengths and limitations of the models.

Summarize main findings:

- Strong Representational Learning
  The macro AUC of ~0.98 shows that the model learned highly discriminative features.

- Moderate Top-1 Accuracy
  An accuracy of ~58% is reasonable for CIFAR-100 given its difficulty and image resolution.
- Balanced Class Performance
  The macro F1 is close to accuracy, indicating that the model performs fairly evenly across classes rather than heavily favoring a few.
- Confusion Matrix and t-SNE
  Visual results confirm good clustering behavior and minimal systematic errors.

Discuss limitations:

  - Low image resolution (32×32) limits the model's ability to learn fine details.
  - Training environment constraints (CPU or limited time) reduced the number of epochs and full fine-tuning potential.
  - Only one model architecture (ResNet-18) was examined in depth; CNN baseline underperformed significantly.
  - No advanced augmentation techniques (CutMix, MixUp, RandAugment) were used.

References:

1. CIFAR-100 Dataset – University of Toronto
   https://www.cs.toronto.edu/~kriz/cifar.html
2. PyTorch Official Documentation (Model Training, torchvision, optimizers)
   https://pytorch.org/docs/stable/index.html
3. Torchvision Model Zoo (ResNet-18 pretrained model)
   https://pytorch.org/vision/stable/models/resnet.html
4. Torchvision CIFAR-100 Dataset Loader
   https://pytorch.org/vision/stable/generated/torchvision.datasets.CIFAR100.html
5. scikit-learn Metrics (accuracy, F1-score, AUC, confusion matrix)
   https://scikit-learn.org/stable/modules/model_evaluation.html
6. scikit-learn t-SNE Documentation
   https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html
7. Wikipedia: Convolutional Neural Network (CNN)
   https://en.wikipedia.org/wiki/Convolutional_neural_network
8. Wikipedia: Residual Neural Network (ResNet)
   https://en.wikipedia.org/wiki/Residual_neural_network
9. Wikipedia: F-score
   https://en.wikipedia.org/wiki/F-score
10. Wikipedia: Receiver Operating Characteristic (ROC)
    https://en.wikipedia.org/wiki/Receiver_operating_characteristic