Assignment -1 Report

Nalla Janardhana Rao - MDS202426
Pranav Pothan - MDS202429
Raja S - MDS202430

Task 1: notMNIST Classification using CNN

Introduction: The notMNIST dataset consists of grayscale images of letters **A–J** in various fonts. The task was to build a **Convolutional Neural Network (CNN)** classifier to recognize these alphabets and to study the effect of permuting image pixels on CNN performance.

Part A: CNN Classification

- The dataset was loaded from Hugging Face (`anubhavmaity/notMNIST`), containing predefined train and test splits.
- Preprocessing steps included normalization of pixel values to $[0, 1]$, adding a channel dimension, and one-hot encoding the labels.
- A CNN architecture was designed with two convolutional layers, max pooling, a fully connected dense layer, and a softmax output layer.
- Training was performed for 10 epochs using Adam optimizer and categorical cross-entropy loss.
- The model achieved a test accuracy of around **93%**, showing strong performance on clean notMNIST data.

Part B: Effect of Pixel Permutation

- To investigate the reliance of CNNs on spatial locality, a fixed random permutation of pixels was applied to every image.
- The CNN architecture remained the same, and training was repeated on the permuted dataset.
- The validation accuracy dropped, converging close to **90%**, which indicates random permutataion of pixels had an effect on accuracy.

Conclusion: The experiment demonstrates that CNNs leverage local pixel neighborhoods to extract spatial features. When the spatial structure is destroyed through permutation, CNNs fail to generalize, unlike fully connected networks that could, in principle, learn on such data. This highlights the importance of preserving spatial coherence in image-based deep learning tasks.

Task 2: Emotion Detection using ResNet-18

Introduction: The goal of this project was to build an emotion detection system using the *Facial Expression Recognition* (FER) dataset from Kaggle. We experimented with two approaches: training a ResNet-18 model from scratch, and fine-tuning a pretrained ResNet-18 on ImageNet. The task involves classifying images into seven emotion categories: **Angry, Disgusted, Fearful, Happy, Neutral, Sad, Surprised**. Since ResNet expects RGB inputs of size 224×224, all images were preprocessed accordingly.

Methodology:

- **Data Preparation:** Dataset was split into train, validation, and test sets. Augmentations such as random crops and horizontal flips were applied to improve generalization.
- **Model 1 (Scratch):** ResNet-18 was trained with randomly initialized weights.
- **Model 2–5 (Fine-tuned):** Pretrained ResNet-18 was fine-tuned under four conditions:
  1. Freeze only `layer1`.
  2. Freeze `layer1 + layer2`.
  3. Freeze `layer1 + layer2 + layer3`.
  4. Freeze all convolutional layers (only FC layer trained).
- **Optimization:** Cross-entropy loss with Adam/SGD optimizers was used, with a step learning rate scheduler.
- **Evaluation:** Models were evaluated using accuracy, confusion matrix, and per-class metrics.

Results:

Model	**Val Acc**	**Test Acc**
ResNet18 (Scratch)	60.4%	59.1%
ResNet18 (FT: Layer1)	**67.0%**	**66.7%**
ResNet18 (FT: Layer1+2)	65.9%	65.5%
ResNet18 (FT: Layer1+2+3)	64.6%	62.9%
ResNet18 (FT: All Frozen)	42.4%	42.4%

Discussion:

- Training from scratch achieved only ∼59% test accuracy, limited by data size.
- Fine-tuning significantly improved performance. The best model was obtained by freezing only `layer1`, which reached 66.7% test accuracy.
- Freezing additional layers (`layer2`, `layer3`) reduced adaptability, leading to lower accuracy.

- Freezing all convolutional layers performed poorly, confirming that deeper layers must adapt to emotion-specific features.
- Confusion matrix analysis showed strong recognition of **Happy** and **Neutral**, while **Sad**, **Fearful**, and **Angry** were frequently misclassified due to visual similarities.

Conclusion: Pretrained models clearly outperform training from scratch. The experiment highlights the importance of fine-tuning depth: partial freezing (early layers) yields the best trade-off between pretrained knowledge and dataset adaptation. Future work may include stronger augmentation, class balancing, and experimenting with deeper models (ResNet-34, ResNet-50) for further gains.