

1. Classifier for Fashion-MNIST: CNN Approach

a) Classification into 3 Classes

The Fashion MNIST dataset contains images from 10 categories, which we relabeled into 3 broad classes: **clothes**, **shoes**, and **others**. The relabeling is as follows: - **Clothes**: T-shirts, pullovers, dresses, coats, shirts. - **Shoes**: Sandals, sneakers, ankle boots. - **Others**: Trousers, bags.

Model Architecture

The model architecture is based on a convolutional neural network (CNN) with the following layers:

- **Input Layer**: 28x28 grayscale image.
- **1st Convolutional Layer**: 32 filters, 5x5 kernel size, padding of 2. (Output: 28x28x32)
- **2nd Convolutional Layer**: 32 filters, 5x5 kernel size, padding of 2. (Output: 28x28x32)
- **Max Pooling Layer**: 2x2 pooling, reducing the spatial dimension to 14x14x32.
- **Dropout Layer**: 25% dropout to prevent overfitting.
- **3rd Convolutional Layer**: 64 filters, 3x3 kernel size, padding of 1. (Output: 14x14x64)
- **4th Convolutional Layer**: 64 filters, 3x3 kernel size, padding of 1. (Output: 14x14x64)
- **Max Pooling Layer**: 2x2 pooling, reducing the spatial dimension to 7x7x64.
- **Dropout Layer**: 25% dropout.
- **Flatten Layer**: Converts the 7x7x64 feature maps into a 1D vector (size: $64 \times 7 \times 7 = 3,136$).
- **Fully Connected Layer**: 512 units, followed by Batch Normalization and ReLU activation.
- **Dropout Layer**: 50% dropout to prevent overfitting.

- **Output Layer:** 3 units corresponding to the 3 classes (clothes, shoes, others), with softmax activation for classification.

The model is trained using **cross-entropy loss** and optimized with the **Adam optimizer**.

After 10 epochs, the model achieved: - **Test Accuracy:** 99.51%

b) Effect of Permuting Pixels

When the pixels of the images were randomly shuffled, the spatial structure was lost, which greatly impacted the CNN's performance: - **Test Accuracy (after permutation):** 99.17%

There was a decrease in Accuracy score when we did the permutation.

This shows that CNNs rely on spatial relationships within images for effective classification.

2. Project Title: Emotion Detection using ResNet

1. Problem Context:

The project focuses on detecting emotions from facial expressions in images using deep learning techniques. This area of research has significant applications in human-computer interaction, surveillance, and mental health assessment.

2. Problem Statement:

How can we accurately classify emotions from images of facial expressions using a convolutional neural network (CNN), specifically leveraging the ResNet architecture.

3. Solution Approach:

a) Training a Classifier from Scratch:

- **Data Collection:** The dataset consists of images categorized into seven emotions: angry, disgusted, fearful, happy, neutral, sad, and surprised. Each emotion is stored in a separate folder.
- **Data Preprocessing:**
 - Images are resized to 224x224 pixels and converted to grayscale to standardize input sizes for the model.
 - **Normalization:** This process involved scaling the pixel values across the full training dataset to achieve a mean of approximately 0.51 and a standard deviation of about 0.20 for the three color channels. Normalization helps in stabilizing the training process and improving model convergence by ensuring the input data maintains a consistent distribution.

- **Model Selection:**
 - A ResNet-18 architecture was chosen for the task, with modifications made to the final layer to accommodate seven output classes corresponding to the different emotions.
- **Training the Model:**
 - The model was trained using the Adam optimizer and CrossEntropy-Loss for 15 epochs. The training loop included forward and backward passes, updating the model weights to minimize the loss function.
- **Performance Evaluation:**
 - The model’s performance was evaluated on a test set, calculating accuracy, loss, and generating a confusion matrix to visualize classification results.
 - **Results from Scratch Training:** The model achieved a final test accuracy of 60.48%

b) Fine-Tuning a Pretrained ResNet-18:

- **Utilization of Pretrained Model:**
 - For improved results, a pretrained ResNet-18 model from PyTorch was fine-tuned on the emotion detection dataset. The final layer was modified to match the seven emotion classes.
- **Layer Freezing Experiment:**
 - We experimented with freezing different layers of the ResNet architecture one by one and assessed the model’s performance for each configuration. This approach allowed us to identify the optimal combination of frozen and unfrozen layers. Ultimately, we selected the configuration that provided the best performance for further training.
- **Training Process:**
 - After identifying the best layer configuration, we trained the model with all layers unfrozen for a reduced number of epochs (e.g., 10 epochs), allowing the model to learn and adjust to the specific nuances of the emotion detection task.
- **Performance Evaluation:**
 - The fine-tuned model was evaluated on the same test set, measuring accuracy and loss.
 - **Results from Fine-Tuning:** The fine-tuned model achieved a test accuracy of 65.10%, indicating significant improvement over the model trained from scratch.

4. Comparison of Results:

- **Performance Metrics:**
 - The classifier trained from scratch achieved a test accuracy of **60.48%**, while the fine-tuned ResNet-18 reached **65.10%** accuracy.

- The confusion matrix for the fine-tuned model showed improved classification performance across various emotions compared to the scratch model.

5. Conclusion:

This project successfully demonstrated the application of the ResNet architecture in emotion detection from facial expressions. The comparison between the scratch-trained model and the fine-tuned model underscored the benefits of using pretrained networks, achieving a notable improvement in classification accuracy.