Mini-Project 2 CS771

Purav Jangir
220837

Priyanshu Maurya
220827

Prerak Agarwal
220818

Burhanuddin Merchant
220300

Lovedeep Sharma
220595

1 Abstract

This project addresses a progressive learning problem using a subset of the CIFAR-10 dataset, where the goal is to sequentially train models while ensuring minimal performance degradation on previously encountered datasets. The methodology employs feature extraction using a pre-trained MobileNet model to generate compact image embeddings, followed by dimensionality reduction via *Principal Component Analysis (PCA)* to a fixed number of components.

2 Task-1

A Learning with prototype-based classifier (*LwP*) is initialized using labeled data from D_1, and class prototypes are iteratively updated using pseudo-labeled data from D_2 to D_{10} under a confidence filtering mechanism to ensure robustness. The pseudo-labeling process leverages the `cosine` distance between features and prototypes, with prototypes updated through a weighted moving average controlled by an interpolation factor α. Feature extraction leverages batch processing to manage computational complexity, while PCA ensures uniform feature representation across datasets. Performance evaluation is conducted on held-out datasets $\hat{D}_1$ to $\hat{D}_{10}$, using accuracy matrices to assess model consistency and progression. The approach ensures that the model f_i adapts to the current dataset D_i while retaining its performance on prior datasets $\hat{D}_1, \ldots, \hat{D}_{i-1}$, minimizing catastrophic forgetting. Experimental results are presented as accuracy matrices, highlighting the efficiency of the progressive learning framework.

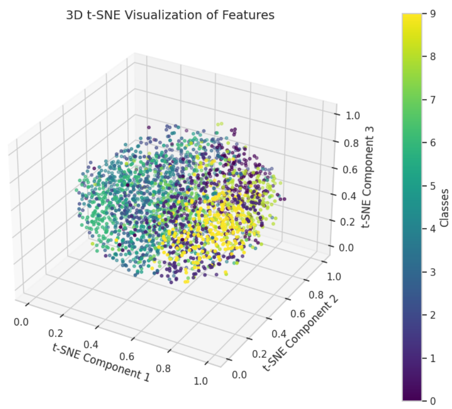
2.1 First (Failed) attempt

Before finalizing the current methodology, we initially attempted an alternative approach for feature extraction and classification. This approach utilized a pre-trained Inception-V3 model to extract features from the input datasets. The extracted high-dimensional features were then reduced to three dimensions using *t-SNE* (t-Distributed Stochastic Neighbor Embedding), a non-linear dimensionality reduction technique commonly used for data visualization. Subsequently, classification was performed using a Learning-with-Prototypes (LwP) classifier, where class labels were assigned based on the *Euclidean distance* between feature points and class prototypes.

Despite the computational rigor, this approach faced significant challenges:

- **Low Accuracy:** The accuracy on the held-out dataset $\hat{D}_1$ was only 35.67%, indicating a suboptimal classification performance.
- **Loss of Information:** Reducing features to three dimensions using *t-SNE* likely resulted in the loss of critical discriminatory information required for effective classification.

- **Sensitivity of Distance Metric:** The use of *Euclidean distance* for prototype learning proved inadequate for capturing the complex nuances of high-dimensional data distributions.



(a) Data points visualized in 3D after applying t-SNE. The clusters indicate class separability, though the overlap suggests suboptimal feature discrimination.

```
[[0.6751913  0.7045543  0.53290254]
 [0.6991954  0.35819012 0.5064613 ]
 [0.40764436 0.562833   0.56953907]
 [0.4169985  0.481464   0.4639294 ]
 [0.35371    0.4753386  0.56045854]
 [0.4052151  0.51935816 0.3697797 ]
 [0.29708028 0.3975441  0.5172005 ]
 [0.52632695 0.4794364  0.43627772]
 [0.730151   0.66971415 0.4598906 ]
 [0.7678766  0.37335816 0.5218274 ]]
```

(b) Prototype vectors in 3D space, computed using Euclidean distance. These represent class centroids but fail to adapt to the complexity of data distributions.

This unsatisfactory performance underscored the need for a more robust pipeline for feature extraction and dimensionality reduction. These insights guided the transition to the improved methodology described in Section below, where *MobileNet* and *PCA* were employed to achieve significantly better results.

2.2 Final Solution

The final solution pipeline consists of the following steps:

- **Feature Extraction:** Pre-trained *MobileNet*, fine-tuned as a feature extractor, processed image data resized to 224×224. The extracted features were global average-pooled for compact representation.
- **Dimensionality Reduction:** PCA was applied to reduce feature dimensions to 256, optimizing computational efficiency while preserving class separability.
- **Classifier Setup:** The LwP classifier initialized prototype vectors for each class based on the mean of corresponding feature clusters. Predictions were made using the Euclidean distance between data points and prototypes.
- **Prototype Update:** During iterative learning, pseudo-labeled data from subsequent datasets refined the prototypes using a confidence threshold ($\geq 90\%$) and an update factor $\alpha = 0.7$.
- **Confidence Filtering**: Ensures that only reliable pseudo-labels are used for prototype updates by filtering out uncertain predictions. The confidence is computed as:

  $$\text{confidence} = 1 - \left(\frac{\text{distances.min(axis=1)}}{\text{distances.max(axis=1)}} \right)$$

  Only points with confidence above the threshold are retained for further processing.

The solution after execution of this pipeline is the following 10x10 matrix, demonstrating the classification accuracy across different datasets after applying the LwP:

```
Accuracy matrix:
[[0.7556 0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.7556 0.742  0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.7556 0.742  0.7364 0.     0.     0.     0.     0.     0.     0.    ]
 [0.7556 0.742  0.7364 0.7512 0.     0.     0.     0.     0.     0.    ]
 [0.7556 0.742  0.7364 0.7512 0.754  0.     0.     0.     0.     0.    ]
 [0.7556 0.742  0.7364 0.7512 0.754  0.7536 0.     0.     0.     0.    ]
 [0.7556 0.742  0.7364 0.7512 0.754  0.7536 0.7344 0.     0.     0.    ]
 [0.7556 0.742  0.7364 0.7512 0.754  0.7536 0.7344 0.754  0.     0.    ]
 [0.7556 0.742  0.7364 0.7512 0.754  0.7536 0.7344 0.754  0.7248 0.    ]
 [0.7556 0.742  0.7364 0.7512 0.754  0.7536 0.7344 0.754  0.7248 0.7604]]
```

Figure 2: 10x10 Matrix Solution on 10 held-out datasets

Methodology	**Accuracy on $\hat{D}_1$**
Inception-V3 + t-SNE + Euclidean distance.	35.67%
MobileNet + Confidence filtering + PCA + Cosine distance.	75.56%

Table 1: Comparison of Results from Different Approaches

3 Task-2

3.1 First (failed) attempt

The pipeline here is mostly the same as `Task-1`, some changes are introduced -

- **Model Architecture and Feature Processing:** The pipeline leverages a pre-trained *MobileNet* as a feature extractor, processing input images resized to 224×224. Features are reduced to 256 dimensions using PCA to enhance computational efficiency. The Learning with Prototype (LwP) classifier initializes class prototypes as centroids of feature clusters and employs these prototypes for centroid-based predictions.
- **Iterative Pseudo-labeling and Domain Adaptation:** Pseudo-labeling with confidence-based filtering is employed to iteratively label unseen datasets. Prototypes are dynamically updated using a weighted regularization strategy that incorporates prior knowledge and newly confident features. Phase 1 (f_1 to f_{10}) processes initial datasets while refining the prototypes iteratively.
- **Extended Training for f_{11} to f_{20}:** Phase 2 extends the training pipeline by reusing prototypes from Phase 1 as prior knowledge for domain adaptation to unseen datasets. The model aligns features using the refined prototypes and evaluates performance across all datasets, ensuring consistency and handling domain shifts effectively. Evaluation is performed for each model on previously encountered datasets to assess long-term adaptation.

The results of `attempt-1` for `Task 2` are shown in the image below:

```
[[0.1268 0.1336 0.1268 0.1168 0.1336 0.1188 0.1196 0.122  0.116  0.1288
  0.1148 0.     0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.1268 0.1336 0.1268 0.1168 0.1336 0.1188 0.1196 0.122  0.116  0.1288
  0.1148 0.0808 0.     0.     0.     0.     0.     0.     0.     0.    ]
 [0.1268 0.1336 0.1268 0.1168 0.1336 0.1188 0.1196 0.122  0.116  0.1288
  0.1148 0.0808 0.114  0.     0.     0.     0.     0.     0.     0.    ]
 [0.1268 0.1336 0.1268 0.1168 0.1336 0.1188 0.1196 0.122  0.116  0.1288
  0.1148 0.0808 0.114  0.1016 0.     0.     0.     0.     0.     0.    ]
 [0.1268 0.1336 0.1268 0.1168 0.1336 0.1188 0.1196 0.122  0.116  0.1288
  0.1148 0.0808 0.114  0.1016 0.122  0.     0.     0.     0.     0.    ]
 [0.1268 0.1336 0.1268 0.1168 0.1336 0.1188 0.1196 0.122  0.116  0.1288
  0.1148 0.0808 0.114  0.1016 0.122  0.0948 0.     0.     0.     0.    ]
 [0.1268 0.1336 0.1268 0.1168 0.1336 0.1188 0.1196 0.122  0.116  0.1288
  0.1148 0.0808 0.114  0.1016 0.122  0.0948 0.0956 0.     0.     0.    ]
 [0.1268 0.1336 0.1268 0.1168 0.1336 0.1188 0.1196 0.122  0.116  0.1288
  0.1148 0.0808 0.114  0.1016 0.122  0.0948 0.0956 0.1008 0.     0.    ]
 [0.1268 0.1336 0.1268 0.1168 0.1336 0.1188 0.1196 0.122  0.116  0.1288
  0.1148 0.0808 0.114  0.1016 0.122  0.0948 0.0956 0.1008 0.1364 0.    ]
 [0.1268 0.1336 0.1268 0.1168 0.1336 0.1188 0.1196 0.122  0.116  0.1288
  0.1148 0.0808 0.114  0.1016 0.122  0.0948 0.0956 0.1008 0.1364 0.1072]]
```

Figure 3: Results for Task 2

3.2 Final Solution

Following changes are made into the improved version of our solution-

- **Domain Adaptation using MMD (Maximum Mean Discrepancy)**: The key modification is the introduction of the `domain_adaptation` function. This function aligns the target domain features (`reduced_features`) to the source domain features (`prev_features`) using Maximum Mean Discrepancy (MMD) as a loss measure. The MMD loss is computed between the source and target feature distributions, and a gradient descent-based adjustment is applied to the target domain features to minimize the distributional gap. Specifically, the `aligned_features` are computed as:

$$\text{aligned\_features} = \text{target\_features} - \lambda_{\text{mmd}} \times \text{mmd\_loss}$$

  where λ_{mmd} is a regularization parameter controlling the strength of the alignment, and `mmd_loss` quantifies the divergence between the source and target distributions.
- **Feature Alignment via Gradient-Based Adjustment**: Once the MMD loss is computed, it is used to adjust the target features, ensuring that they are closer to the source domain features. This adjustment makes the target features more comparable to the source features, thereby improving transfer learning capabilities. The alignment process ensures that the learned representations for new classes are consistent with those learned previously, reducing domain shift issues.

The solution after execution of this version-

```
Accuracy Matrix :
[[60.18 62.6  64.61 65.59 65.35 63.99 61.91 60.3  60.35 60.35 60.33
   0.    0.    0.    0.    0.    0.    0.    0.    0.  ]
 [60.44 62.6  64.61 65.59 65.35 63.99 61.91 60.36 60.38 60.09 60.21
  60.18  0.    0.    0.    0.    0.    0.    0.    0.  ]
 [60.35 62.6  64.61 65.59 65.35 63.99 61.91 60.39 60.19 60.42 60.38
  60.31 61.    0.    0.    0.    0.    0.    0.    0.  ]
 [60.37 62.6  64.61 65.59 65.35 63.99 61.91 60.42 60.33 60.26 60.31
  60.45 61.   63.68  0.    0.    0.    0.    0.    0.  ]
 [60.41 62.6  64.61 65.59 65.35 63.99 61.91 60.23 60.19 60.49 60.46
  60.04 61.   63.68 66.08  0.    0.    0.    0.    0.  ]
 [60.33 62.6  64.61 65.59 65.35 63.99 61.91 60.1  60.25 60.2  60.01
  60.13 61.   63.68 66.08 67.69  0.    0.    0.    0.  ]
 [60.23 62.6  64.61 65.59 65.35 63.99 61.91 60.44 60.44 60.39 60.22
  60.25 61.   63.68 66.08 67.69 68.15  0.    0.    0.  ]
 [60.18 62.6  64.61 65.59 65.35 63.99 61.91 60.35 60.2  60.39 60.34
  60.04 61.   63.68 66.08 67.69 68.15 67.39  0.    0.  ]
 [60.24 62.6  64.61 65.59 65.35 63.99 61.91 60.25 60.39 60.21 60.46
  60.31 61.   63.68 66.08 67.69 68.15 67.39 65.66  0.  ]
 [60.02 62.6  64.61 65.59 65.35 63.99 61.91 60.17 60.46 60.49 60.37
  60.37 61.   63.68 66.08 67.69 68.15 67.39 65.66 63.42]]
```

Figure 4: Results for Task 2

Methodology	**Accuracy on $\hat{D}_1$**
Without Feature Alignment	12.68%
After Domain Adaptation using MMD	60.18%

Table 2: Comparison of Results from Different Approaches

4 Video Presentation

Our team has created a detailed presentation on the research paper titled *"Lifelong Domain Adaptation via Consolidated Internal Distribution"*. The video provides an in-depth overview of the methodology, experiments, and insights discussed in the paper. You can watch the presentation at the following link:

GROUP-24 VIDEO PRESENTATION

Acknowledgments

All the links are provided at the introduction of the concept.