Indian Institute of Technology Kanpur

INTRODUCTION TO MACHINE LEARNING (CS771)

MINI-PROJECT 2 REPORT

Students Name	Student ID
Param Soni	220752
Samrat Patil	220953
Mohd Nasar Siddiqui	220661
Abhishek Kumar	220044
Kartik	220503

Instructor:

Dr. Piyush Rai

Submission Date : 26/11/2024

1 Introduction

This research aims to develop a resilient **Learning with Prototypes (LwP)** classifier capable of addressing key challenges associated with pseudo-label generation and domain adaptation. The proposed iterative framework focuses on training models across a sequence of datasets with shared or marginally shifted distributions, while effectively mitigating the phenomenon of **catastrophic forgetting**.

The study tackles the complexities of incremental learning on multiple subsets of the **CIFAR-10 dataset**, employing the LwP methodology to iteratively train a series of models, $f_1, f_2, \ldots, f_{20}$, on corresponding datasets, $D_1, D_2, \ldots, D_{20}$. The primary objective is to optimize classification accuracy on held-out labeled subsets of each dataset while simultaneously minimizing performance deterioration on earlier datasets and ensuring robust generalization to accommodate distributional shifts (*domain shift*).

To achieve these goals, the framework integrates a pre-trained **ResNet-50** for feature extraction and draws on methodologies outlined in the *Déjà Vu: Continual Model Generalization for Unseen Domains* paper. By leveraging these concepts, the models are systematically refined to preserve performance across previously encountered datasets while maintaining consistent generalization capabilities under varying distributional scenarios.

2 Problem Statement

Problem Overview

The problem focuses on a sequence of 20 datasets $\{D_1, D_2, \ldots, D_{20}\}$, where D_1 is fully labeled and serves as the initial training set, while D_2 to D_{20} are unlabeled. Among these, D_2 to D_{10} share the same distribution as D_1, whereas D_{11} to D_{20} belong to different but related distributions.

Objectives

The main goals are to generate reliable pseudo-labels for the unlabeled datasets $\{D_2, \ldots, D_{20}\}$, iteratively train models $\{f_1, f_2, \ldots, f_{10}\}$ on these datasets while ensuring minimal performance degradation on previously trained datasets, and evaluate the models on previously seen datasets using held-out labeled sets.

Constraints

The solution is constrained by the restriction against using neural networks or advanced models for training, requiring reliance solely on Learning with Prototypes (LwP) methodologies. Pretrained neural networks may only be used for feature extraction.

Challenges

Pseudo-Labeling Unlabeled Data

- Datasets $\{D_2, \ldots, D_{20}\}$ lack labels, requiring the creation of pseudo-labels.

- The model's ability to generalize across datasets is highly dependent on the accuracy of these pseudo-labels.
- A key challenge is finding the right balance between diversity and pseudo-label confidence.

Domain Shift

Datasets $D_{11}, \ldots, D_{20}$ exhibit statistical distributions that diverge from those of $D_1, \ldots, D_{10}$, representing a significant domain shift. This necessitates the model's ability to adapt effectively to the new distributions while maintaining its performance integrity on the original datasets. This scenario underscores the critical importance of robust domain adaptation methodologies to address the challenges posed by such distributional discrepancies.

Managing Catastrophic Forgetting

- Performance deterioration may occur as the model forgets information from earlier datasets while iteratively training on new ones.
- Mechanisms such as memory-based approaches and prototype retention should be employed to manage this issue.

3 Methodology

This section outlines the systematic approach employed to tackle the challenge of incrementally updating models across sequential datasets. The proposed methodology is designed to effectively mitigate catastrophic forgetting while simultaneously enabling adaptation to evolving domain shifts.

Feature Extraction

- A pre-trained ResNet-50 model is employed for feature extraction:
  - For each input image x_i, a fixed-size feature vector $z_i \in \mathbb{R}^{2048}$ is obtained by removing the last fully connected (classification) layer.
  - The extracted features are then used for prototype construction and pseudo-label generation.
- The feature extraction procedure can be expressed mathematically as:

$$z_i = f_{\text{resnet}}(x_i; \theta_{\text{fixed}})$$

  where θ_{fixed} represents the frozen parameters of the pre-trained network and f_{resnet} is the ResNet-50 backbone.

Mathematically, the feature extraction process can be represented as:

$$\mathbf{z}i = f\text{resnet}(x_i; \theta_{\text{fixed}})$$

where f_{resnet} is the ResNet-50 backbone, and θ_{fixed} represents the frozen parameters of the pre-trained network.

Constructing Prototype for Model

Class prototypes c_k are derived by aggregating the feature representations of labeled data points, mathematically expressed as:

$$c_k = \frac{1}{|D_k|} \sum_{i \in D_k} z_i, \quad \forall k \in \{1, \ldots, C\}$$

Here, C represents the total number of classes, and D_k denotes the set of feature vectors associated with class k.

These prototypes act as the canonical representations of their respective classes and are instrumental in facilitating pseudo-label generation while aligning the distributions of incoming data with previously learned representations.

Cosine Similarity for Pseudo-Labeling

Cosine similarity is utilized to generate pseudo-labels for the unlabeled datasets $\{D_2, \ldots, D_{20}\}$. Specifically, pseudo-labels are assigned based on the similarity between the feature representation of each unlabeled sample z_i and the class prototypes c_k. For a given feature vector z_i, the cosine similarity with all class prototypes is computed. The pseudo-label $\hat{y}_i$ is then attributed to the class k_1 corresponding to the prototype with the highest similarity score:

$$\hat{y}_i = \arg\max_k \text{Sim}(z_i, c_k),$$

where the similarity measure is mathematically defined as:

$$\text{Sim}(z_i, c_k) = \frac{z_i \cdot c_k}{\|z_i\| \, \|c_k\|}.$$

High Confidence Selection

For reliable prototype updates, the following procedure is used: First, the difference (confidence) between the highest and second-highest similarity scores is determined. Next, the sample fraction with the highest confidence is selected. In this approach, a threshold of 0.8 is chosen as the top fraction for selection.

We chose top fraction as **0.8** .

Prototype Contrastive Alignment

To address the domain shift present in the datasets $\{D_{11}, \ldots, D_{20}\}$, the Prototype Contrastive Alignment (PCA) approach is employed. This methodology ensures that the feature distributions of the new data are effectively aligned with the existing class prototypes.

The PCA mechanism updates the prototypes iteratively according to the following formulation:

$$c_k^{\text{aligned}} = \alpha c_k^{\text{prev}} + (1 - \alpha) c_k^{\text{curr}}, \quad \forall k \in \mathcal{C},$$

where:

- c_k^{prev} denotes the prototype of class k from the preceding iteration.
- c_k^{curr} represents the prototype of class k computed from the current dataset.
- c_k^{aligned} corresponds to the updated (aligned) prototype for class k.
- $\alpha \in [0, 1]$ is a hyperparameter that governs the balance between the previous and current prototypes.
- $\mathcal{C}$ is the set of all classes present in the dataset.

In the case where a class k is present exclusively in the current dataset and absent from the previous iteration, the prototype is initialized as:

$$c_k^{\text{aligned}} = c_k^{\text{curr}}.$$

For the experiments conducted, the value of α was set to 0.5.

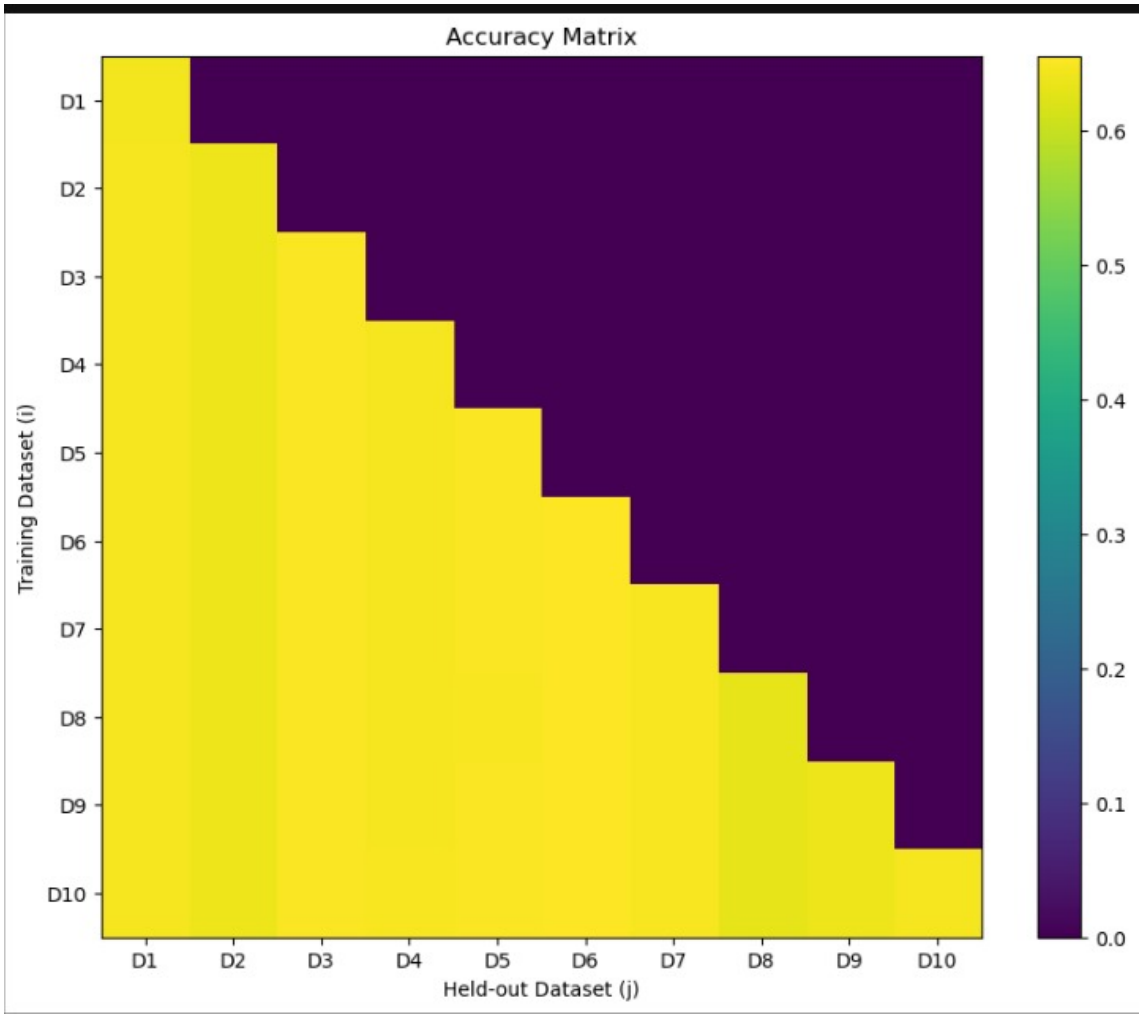
Iterative Training and Evaluation

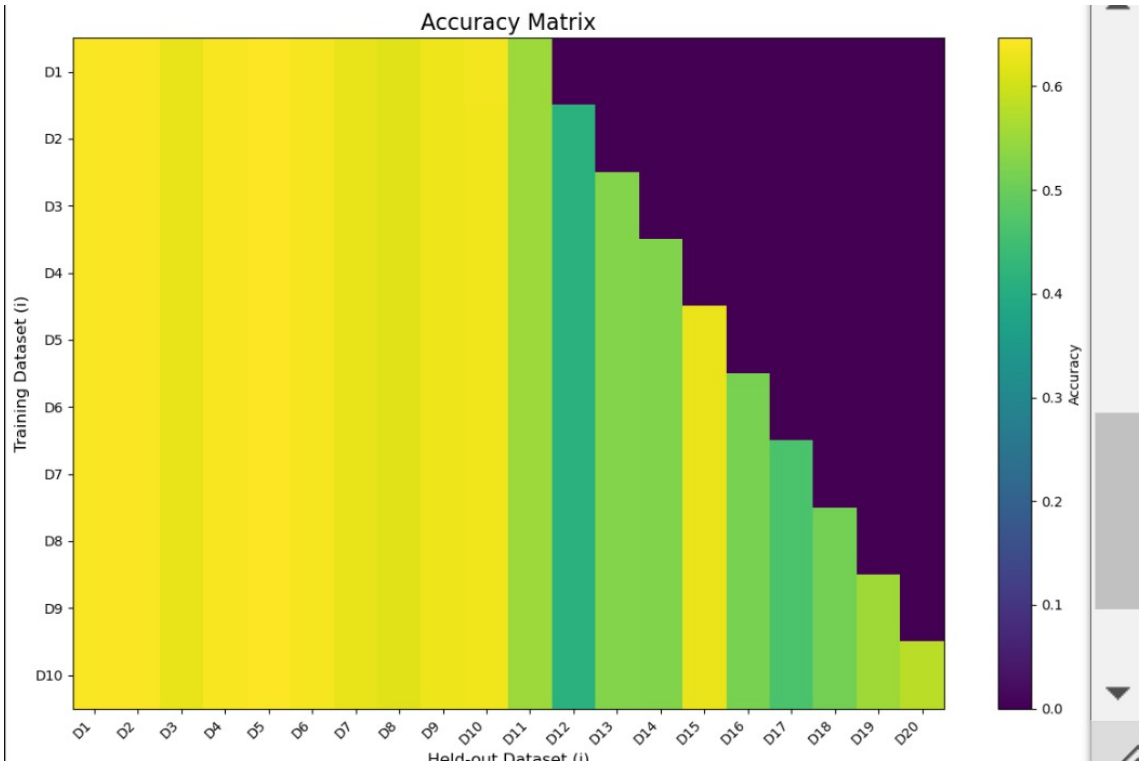
Every unlabeled dataset, from D2 to D20, was subjected to the same methodology. No differentiation was drawn between Tasks 1 and 2, as the prototype-based methodology and pseudo-label creation stayed the same. This consistency makes sure that the pipeline for pseudo-labeling and model updates is methodical and repeatable. Performance is assessed on both recent and historical datasets using held-out labeled sets, and the models are iteratively updated across datasets.

Tried Approaches and accuracies(not final)

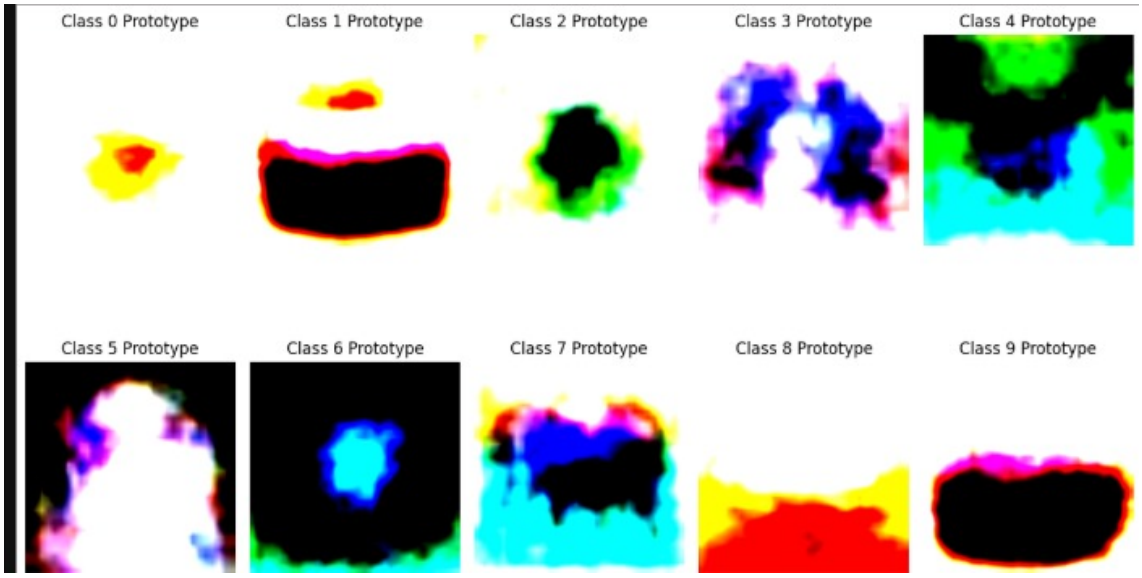
GMM Model

4 Observation and Analysis

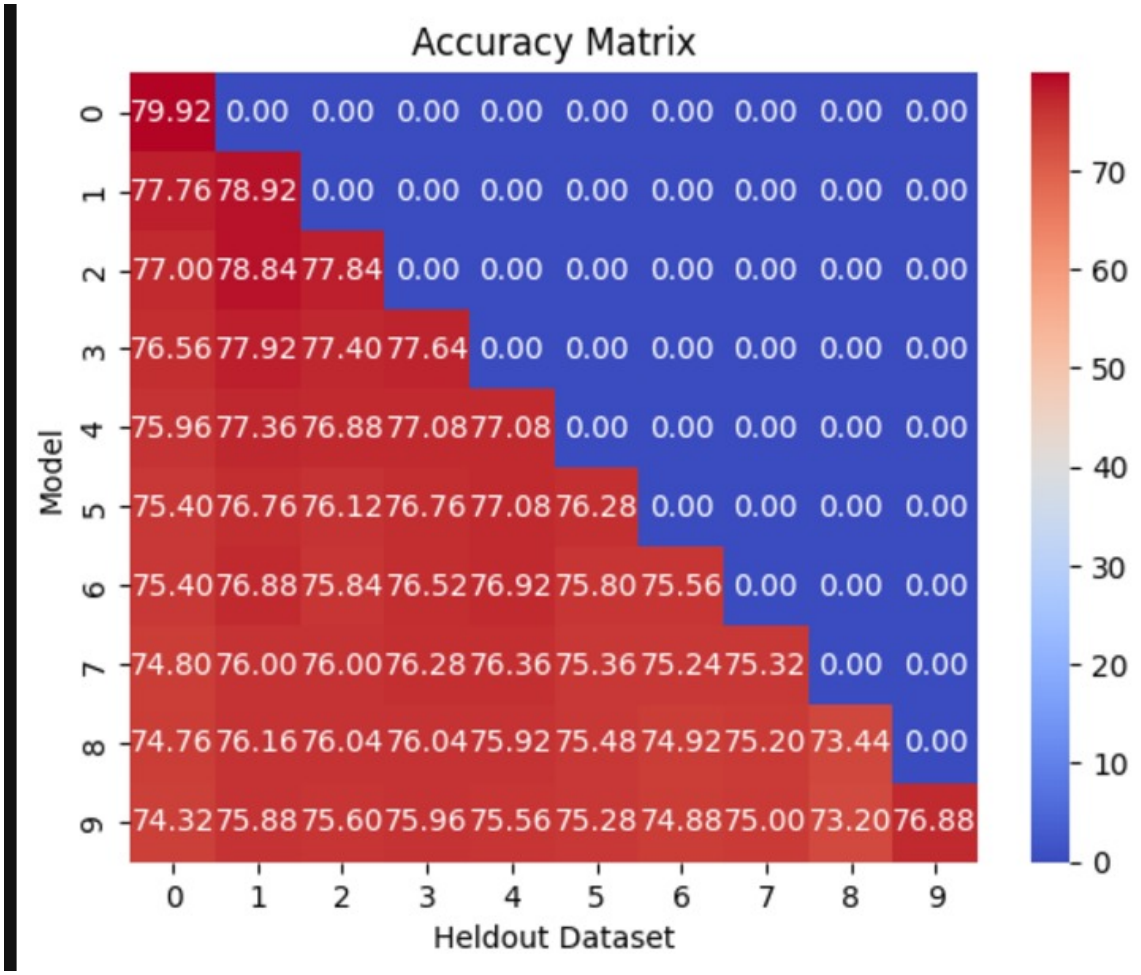




4.1 class prototype visualized



5 Observation and Analysis



Accuracy Matrix (Tabular View)

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20
Model 1	79.92	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 2	78.76	79.96	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 3	77.76	79.4	78.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 4	76.76	78.4	77.8	78.28	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 5	76.04	77.64	77.28	77.44	77.52	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 6	75.72	77.0	76.24	77.08	77.16	76.48	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 7	75.6	77.04	76.04	76.72	77.32	76.24	75.68	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 8	75.16	76.4	76.08	76.56	76.68	75.92	75.28	75.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 9	74.8	76.24	75.96	76.08	76.28	75.8	75.08	75.44	73.68	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 10	74.6	76.16	75.64	76.08	75.84	75.72	75.24	75.08	73.44	77.04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 11	71.96	72.56	73.88	73.72	72.28	73.12	72.64	72.88	70.8	74.6	60.04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 12	67.2	67.2	68.8	68.84	68.4	67.96	67.8	68.24	66.16	70.84	56.24	51.96	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 13	66.12	65.56	67.96	68.24	67.16	66.96	66.44	67.08	65.24	69.84	55.44	51.56	60.36	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Model 14	64.72	64.76	67.88	66.56	66.08	65.88	66.04	66.92	64.64	68.32	54.32	51.2	59.32	62.52	0.0	0.0	0.0	0.0	0.0	0.0
Model 15	66.72	66.84	69.08	68.52	67.68	67.28	67.08	68.04	66.16	70.0	55.72	51.16	60.8	63.24	67.24	0.0	0.0	0.0	0.0	0.0
Model 16	64.88	65.68	67.72	67.56	66.68	65.84	66.2	66.6	64.6	68.88	54.56	50.24	59.64	62.64	66.08	57.84	0.0	0.0	0.0	0.0
Model 17	64.72	63.76	67.72	66.96	65.56	65.24	65.6	66.04	64.08	68.0	54.84	50.64	59.76	62.36	65.16	57.36	58.24	0.0	0.0	0.0
Model 18	63.24	63.08	65.64	64.64	64.36	63.92	64.52	63.88	62.76	66.4	53.12	50.16	58.52	61.56	63.88	56.52	57.36	57.76	0.0	0.0
Model 19	62.64	62.0	63.92	64.64	63.56	62.84	63.16	63.36	62.0	66.04	51.64	49.04	57.76	59.56	63.68	55.28	55.72	57.32	53.28	0.0
Model 20	63.84	63.4	65.96	65.32	65.4	64.32	64.4	64.88	63.08	67.4	53.64	49.76	58.92	60.16	65.28	56.44	56.96	57.56	52.76	61.44

5.1 Accuracy Trends Across Datasets

Because of their similar distribution, the model's performance on datasets D1 through D10 seems to be consistent. Moving on to datasets D11 to D20, however, results in a discernible decline in accuracy, most likely due to a domain shift (i.e., a change in the data distribution).

5.2 Effects of Catastrophic Forgetting

- As the model trains on subsequent datasets, its performance on earlier datasets tends to degrade.
- This phenomenon, referred to as *catastrophic forgetting*, occurs when the model overwrites representations of previously encountered datasets while learning new ones.

5.3 Challenges in Domain Adaptation

- The model encounters significant challenges in adapting to the domain shift between D_1–D_{10} and D_{11}–D_{20}.
- This reveals the limited generalization capacity of the features extracted using ResNet-50 and the prototype-based learning framework.

5.4 Dataset-Specific Observations

- Certain datasets (e.g., D_{12}) exhibit consistently lower performance metrics.
- This may indicate:
  - Higher intrinsic complexity within these datasets, or
  - Inadequate compatibility with the feature representations produced by ResNet-50.

5.5 Comprehensive Performance Analysis

- The average accuracy across all datasets provides a broader assessment of the method's capabilities and limitations.
- If the performance on D_1–D_{10} significantly exceeds that of D_{11}

6 Task 2 link

https://youtu.be/wVz$_0$ $CCYY4Q$

7 References

- **Pandas Documentation**
  https://pandas.pydata.org/
- **Numpy Documentation**
  https://numpy.org/
- **Accuracy score Documentation**
  https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_sco
- **Pytorch Documentation**
  https://pytorch.org/docs/stable/index.html
  https://pytorch.org/vision/stable/
  https://pytorch.org/docs/stable/nn.html
- **Pickle Documentation**
  https://docs.python.org/3/library/pickle.html
- **Resnet50 Documentation**
  https://pytorch.org/vision/0.18/models/generated/torchvision.models.resnet50
- **Deja Vu : Continual Model Generalization for Unseen Domains**
  https://arxiv.org/pdf/2301.10418