CS 771: Intro to Machine Learning
Mini Project 2: Implementation Report

Fall Semester 2024

Kanishk Goyal - 220498
Aditya Modi - 220071
Naman Kumar Jaiswal - 220687
Piyush Singh - 220769

Introduction

In this report, we present the implementation details and results of a Learning with Prototypes (LwP) classifier, applied to a sequence of datasets with evolving distributions. The task involves leveraging labeled data from the initial dataset to iteratively label and refine predictions for subsequent unlabeled datasets, ensuring that the model adapts effectively without significant performance degradation on earlier datasets. The objectives for our Task 1 are as follows:

1. **Train a LwP Chain:** Train the initial LwP model on $D1$ and iteratively update it using predicted labels from $D2$ to $D10$, without reusing previous datasets.
2. **Evaluate on Held-Out Datasets:** Test the updated models $f1$ to $f10$ on held-out datasets $D1$ to $D10$ and report accuracies in a matrix.
3. **Maintain Performance:** Updates to the model mustn't degrade performance on prior datasets.

For the task 2 we will be dealing with 10 more datasets which are extracted from 10 different distributions, not the same as that from 1 to 10. The objective for Task 2 is to **Consider Distribution Differences:** Account for potential distribution shifts between datasets $D11$ to $D20$ during model updates.

Dataset Description & Analysis

The dataset consists of subsets from the **CIFAR-10 image classification dataset**, which contains 10 classes. Each dataset has 2500 images, where each image is represented as a 32×32 matrix of pixels with RGB values from 0 to 255.

- **D1** is the only dataset with both features and labels, while the remaining datasets (**D2 to D20**) only contain features and no labels. For model accuracy computation, the held-out datasets $\tilde{D}_i$ $(i = 1 \dots 20)$ are used for evaluation, but they must not be used during training or cross-validation.

The first 10 datasets (**D1 to D10**) share the same input distribution, meaning the features of these datasets are similar. However, the last 10 datasets (**D11 to D20**) originate from different distributions, differing from both each other and the first 10 datasets. They do share a slight generalised similarity.

Since the raw images are presented as 32×32 pixel matrices, simple flattening of the images won't provide effective feature extraction. More advanced methods, such as using pre-trained neural networks for feature extraction or applying kernels, are necessary to capture meaningful patterns in the images.

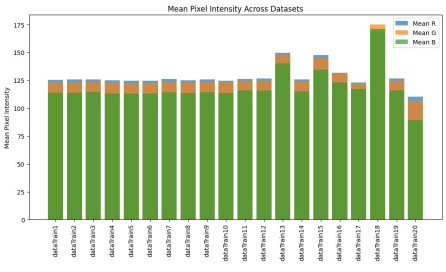

Figure 1: Distribution of Mean RGB values in all the training datasets

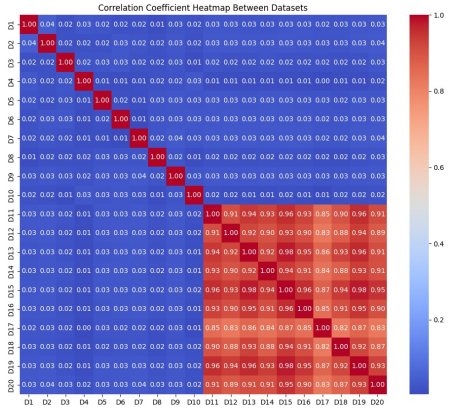

Figure 2: Coefficient of Correlation between Datasets, the more the correlation, the more similar is the parent distribution

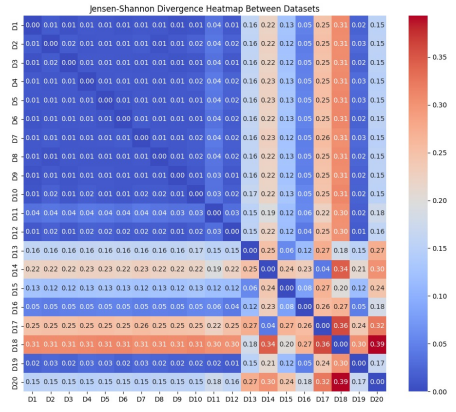

Figure 3: Jensen Shannon Divergence between Datasets, more the divergence, less is the similarity

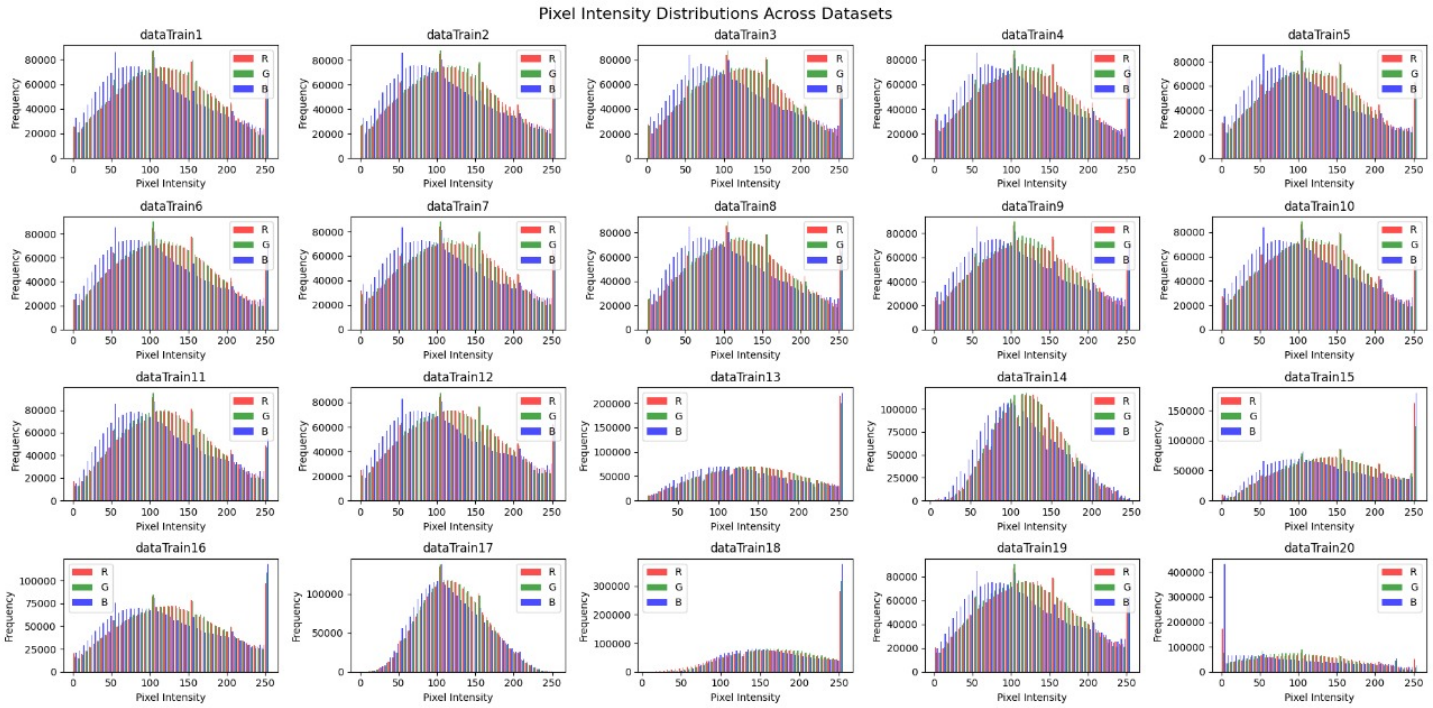

Figure 4: Distribution of frequency of each of the RGB values over the Pixel Intensity for all the datasets

Problem 1 Task 1: Sequential LwP Chain

For task 1, we worked through majorly 2 approaches in order to train an efficient LwP Chain on the data sets - the first one was using ResNet-18 as a feature extractor and other was using a ViT transform to convert the images into another normalized dataset, ready for use in LwP.

ResNet-18 as a Feature Extractor:

ResNet-18, a deep convolutional network from the ResNet family, is known for its residual learning framework, enabling very deep networks to mitigate the vanishing gradient problem. With 18 layers of learnable weights, ResNet-18 offers hierarchical feature representation and computational efficiency. We used a pre-trained ResNet-18 on the ImageNet dataset to extract features, which were then refined using feature selection methods. The final feature set was used to sequentially train the LwP chain. However, this approach yielded only 54% accuracy, likely due to the resolution mismatch and lack of fine-grained details in CIFAR-10 compared to ImageNet.

ViT Transformer for Feature Transformation (Best Approach):

he Vision Transformer (ViT) adapts the Transformer architecture for image tasks by dividing input images into fixed-size patches, flattening these patches into vector embeddings, and processing them as a sequence along with positional encodings to preserve spatial information. This architecture allows ViT to model global relationships between patches using multi-head self-attention, enabling it to effectively capture patterns and dependencies that span across the entire image. This global modeling capability helped ViT preserve the delicate and distributed features of CIFAR-10, even after transformation into a flat dataset. After normalization, the transformed data was directly used to train LwP chains.

Using this method, the LwP chain achieved $\tilde{\mathbf{95\%}}$ average accuracy, with minimal degradation when evaluating earlier f_i models on later D_j datasets. ViT's superior performance over ResNet-18 underscores its flexibility in handling the simpler structure of CIFAR-10 while leveraging its ability to model long-range dependencies across image patches.

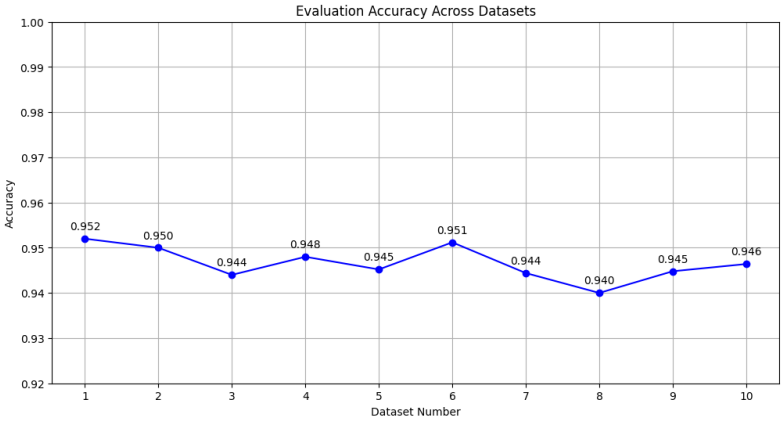

Figure 5: The Accuracy recorded on running f_i on D_i $\forall i \in \{1, 2, \cdots 10\}$

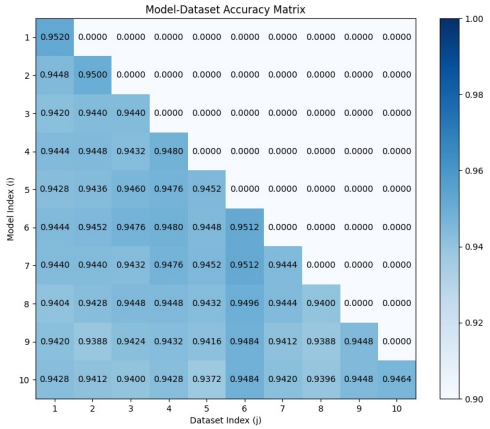

Figure 6: Accuracy on running f_i on D_j $\forall i \in \{1, 2, \cdots 10\}$ $\forall j \in \{1, 2, \cdots i\}$

Problem 1 Task 2: Different Data Distributions

For task 2 also, we mainly used 2 approaches - ViT transformer and implementing Augmentation by introducing noise into the dataset, forcing the bias of our model to go down and make the model try to capture even more general trends.

Adding Noise

We referred to the "Deja Vu: Continual Model Generalization for Unseen Domains" paper, and tried to add small noise onto the data and then tranform and run models. Though we ran into the conclusion that adding noise did reduce the accuracies on the 1 to 10 datasets but the increase we saw for 11 to 20 was not significant enough to prefer this method over our next approach.

ViT Transformer

Since, ViT Transform also has a normalization layer, which increases the similarity between the datasets and makes it also a very reasonable model to run for datasets from 1 to 20. We saw an approximate accuracy of $\tilde{8}8\%$.

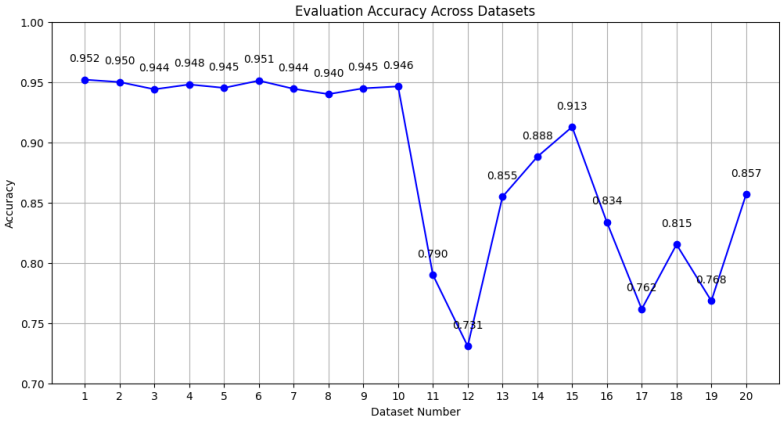

Figure 7: The Accuracy recorded on running f_i on D_i $\forall i \in \{1, 2, \cdots 20\}$

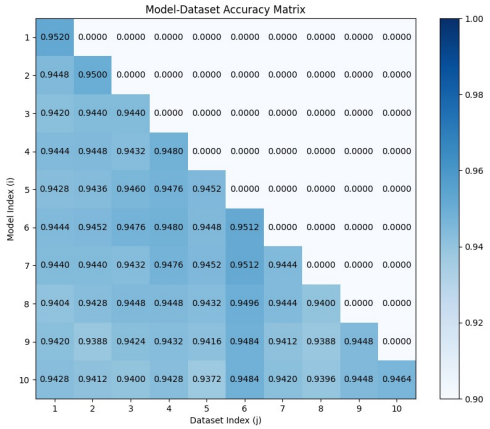

Figure 8: Accuracy on running f_i on D_j $\forall i \in \{1, 2, \cdots 20\} \forall j \in \{1, 2, \cdots i\}$

Problem 2:

The link for our presentation on Lifelong Domain Adaptation via Consolidated Internal Distribution (NeurIPS 2021) $\cdots$ **Here**