CS 499 - Project Report Submission

[Bhavini Korthi - 20110039, Kareena Beniwal- 20110095, Priya Gupta - 20110147]

SSL - Self-Supervised Learning

Problem Statement:
Reviewing research conducted on image-based SSL and coreset search for SSL methods.

LITERATURE REVIEW 1: https://arxiv.org/abs/2305.13689
What and Why SSL?
Shortcomings of Supervised Learning:
1. Not all datasets come with an abundance of labeled training data
2. Robust feature extraction can become a challenge when data is scarce.

The idea behind SSL: devise an experimental setting in which the task that provides the supervisory signal can be solved without human annotation and then train DNNs to solve it.

Generative SSL:
The task is to build appropriate distributions over a collection of data while operating in the pixel space.

Discriminative SSL:
The task is to learn good representations of the data in order to perform a specified pretext task.

Pretext tasks: Pretext tasks enable the model to learn useful representations from unlabeled data, which can then be fine-tuned for specific tasks with limited labeled data. Ex: Image colorization, Inpainting, Geometric transformations, masked image modeling etc.

Loss functions to train SSL frameworks:
Cross Entropy Loss:

$$\mathcal{L}_{\text{CE}}(\hat{\boldsymbol{y}}, t) = -\log \frac{\exp(\hat{\boldsymbol{y}}_t)}{\sum_{c=0}^{C} \exp(\hat{\boldsymbol{y}}_c)}$$

Cosine Similarity:

$$\text{sim}(\boldsymbol{v}_1, \boldsymbol{v}_2) = \frac{\boldsymbol{v}_1 \cdot \boldsymbol{v}_2}{\|\boldsymbol{v}_1\| \|\boldsymbol{v}_2\|}$$

LITERATURE REVIEW 2: https://ieeexplore.ieee.org/document/1640964
Dimensionality Reduction by Learning an Invariant Mapping (DrLIM)
DrLIM is based on mapping complex data into lower dimensional space while making sure that similar points are mapped to nearby points. The similarity or dissimilarity between two data points is defined with the help of prior knowledge about their relationship or by manual labeling. It aims to maintain neighborhood relationships without the necessity of any specific distance measures. DrLIM can even learn mappings that stay the same under certain transformations of the data, and because of this flexibility, it performs well compared to other techniques like LLE, PCA, etc.

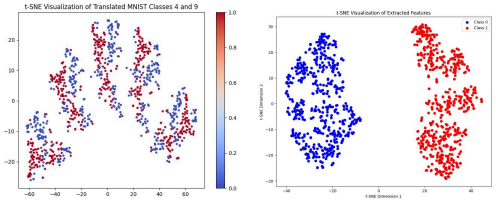
The input vector $I = \{\overline{X}_1,..,\overline{X}_P\}$, where $\overline{X}_i \in \mathbb{R}^D$, $\forall\, i \in 1,...,n$, A parametric function G_W: $\mathbb{R}^D \rightarrow \mathbb{R}^d$ where $d << D$ is required. G_W (suppose CNN network) is trained using the contrastive loss function following the algorithm below.

Algorithm:
(i) Based on the prior relations in the data (or labeling), pair all the data points either *similar* ($y_{ij} = 0$) or *dissimilar* ($y_{ij} = 1$).
(ii) Iterate until convergence;
For each pair $(\overline{X}_i, \overline{X}_j)$ in the training dataset, update the parameter W to

A. Decrease $D_W = ||G_W(\overline{X}_i) - G_W(\overline{X}_j)||_2$, if $y_{ij} = 0$ (*similar*).
B. Increase $D_W = ||G_W(\overline{X}_i) - G_W(\overline{X}_j)||_2$, if $y_{ij} = 1$ (*dissimilar*).

Training: A Siamese architecture containing two copies of the same function G_W and sharing the same parameter W. The input is given in the form of a pair of images $(\overline{X}_i, \overline{X}_j)$. A cost function is employed to measure the distance between the two outputs of G_W and the parameter W is updated using stochastic gradient descent with back-propagation.

The described algorithm is tested on the MNIST dataset considering only the labels 4 and 9. Horizontal translations by -6, -3, 3, and 6 pixels are applied to the data and added to the original dataset.



Implementation: Github Repo Link
The first image represents the clustering of the data containing translated images, and 10 clusters can be observed (5 translations × 2 labels), whereas the second image is the representation of the dataset after the dimensionality reduction, and now we can observe only two clusters (2 classes).

LITERATURE REVIEW 3: https://arxiv.org/abs/1406.6909
Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks
The paper introduces a novel approach to unsupervised feature learning using Convolutional Neural Networks (CNNs) that does not rely on any labeled data but rather makes use of a surrogate task automatically generated from unlabeled images.
The representation learned by the Exemplar-CNN is discriminative while also invariant to some typical transformations, leading to improved performance in classification and matching tasks.
Learning Algorithm: The learning algorithm involves training a CNN to discriminate between surrogate classes formed by sets of transformed image patches. The loss function $L(X)$ is defined as the sum of losses over transformed samples T_{xi} with their corresponding surrogate true label i. The loss function is formulated as

$$L(X) = \sum_{x_i \in X} \sum_{T \in T_i} l\left(i, Tx_i\right)$$

where $l\left(i, T_{xi}\right)$represents the loss on the transformed sample T_{xi} with the true label i. The multinomial negative log-likelihood of the network output is optimized, given by:

$$l\left(i, Tx_i\right) = M\left(e_i, f\left(Tx_i\right)\right)$$

$$M(y, f) = -y \cdot \log f = -\sum_k y_k \log f_k$$

where $f(.)$ computes the output layer values of the CNN given the input data, $e_i$is the i-th standard basis vector, and $f_k$represents the output score for the class k.
By considering the random vector of transformation parameters α, the activations of the network, and the softmax output, the objective function takes the form:

$$\sum_{x_i \in X} E_a \left[-\left(e_i, h\left(T_\alpha x_i\right)\right) + \log \left\| exp\left(h\left(T_\alpha x_i\right)\right)\right\|_1 \right]$$

Experiment and results: The experiments conducted evaluate the performance of the proposed method in both classification and descriptor-matching tasks. The results demonstrate significant improvements in classification accuracy compared to previous unsupervised methods across different datasets. Moreover, the learned features outperform SIFT features in descriptor-matching tasks, particularly when trained with blur transformations.