Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 00 (2025) 000–000
Procedia Computer Science
www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering

Bridging Domain Gaps with CORAL-Enhanced CNNs: A Lightweight Approach to Unsupervised Adaptation

Ansh Bajpai, Nishant Shah, G Gopakumar

Department of Computer Science and Engineering
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Amritapuri, Kollam, India

Abstract

Deep learning models often struggle when applied to data distributions that differ from their original training environment—a challenge known as domain shift. In this study, we explore practical strategies for domain adaptation to address this issue, evaluating their effectiveness across three distinct datasets: MNIST, MNIST-M and Office-31. We begin by establishing baseline performance using traditional classifiers such as Support Vector Machines (SVMs). Next, we assess Convolutional Neural Networks (CNNs) enhanced with Correlation Alignment (CORAL) loss, a method designed to align the feature representations between both domains (i.e. source and target). To further boost model generalization, we incorporate training with data augmentation techniques. Our experiments reveal that the combination of CORAL-based alignment and data augmentation consistently improves cross-domain classification accuracy, benefiting both digit recognition and real-world object classification. These findings highlight the value of aligning statistical characteristics in transfer learning and demonstrate the robustness of CORAL in domain adaptation.

Keywords: Domain Adaptation, Domain Shift, CORrelation ALignment, Maximum Mean Discrepancy, Domain Alignment, Convolutional Neural Network

1. Introduction

Deep learning, especially through CNNs, has led to major breakthroughs in computer vision; from image classification to object detection and scene parsing [5]. Despite their success, these models frequently depend on an uncertain assumption that the distribution of the training data closely resembles the real-world data they will ultimately encounter. Supervised machine learning approaches work well only when the test and training data are from the same

E-mail address: gopakumarg@am.amrita.edu

distribution [8]. In practice, that alignment is rare. A slight variation in lighting, background, style or sensor quality can lead to a domain shift, causing a significant degradation in performance when models are deployed in unseen target environments.

Domain Adaptation (DA) offers a pragmatic solution. It allows models trained on a labeled source domain to generalize effectively to a target domain, often unlabeled. In unsupervised domain adaptation (UDA), the target domain lacks labeled data making the task considerably more difficult and applicable to real-world scenarios. This paper focuses on UDA techniques that use both synthetic and real-world datasets.

We start by analyzing MNIST and MNIST-M: a canonical example of domain shift in a simplified context. These have alterations that are enough to derail conventional models such as SVMs and Random Forests, providing a baseline for understanding the effects of distribution misalignment, the primary cause being that MNIST is grayscale while MNIST-M is colored. We also investigated domain adaptation within a practical context by assessing models using the Office-31 dataset [3], which comprises 31 common object categories. These domains differ in resolution, lighting and style, allowing us to examine model generalization across various transfer tasks, including Amazon vs. Webcam and DSLR vs. Amazon.

In order to address domain shift, we implement CNN-based feature extraction enhanced with CORAL loss, which minimizes the difference in second-order statistics (feature covariances) between the source and target domains.

We further enhance generalization by augmenting source data using geometric and photometric transformations. This promotes domain-invariant feature learning and prepares the model for out-of-distribution variations. When we used it alongside CORAL, augmentation significantly improves adaptation.

Our work makes the following contributions:

- **Baseline Study:** We establish baseline results using traditional machine learning models on both MNIST and MNIST-M to quantify the impact of domain shift.
- **CNN + CORAL Framework:** We implement a CNN-based pipeline with CORAL loss for unsupervised domain adaptation across both synthetic and real-world domains.
- **Hybrid Strategy:** We show that combining data augmentation with CORAL-based domain alignment leads to superior performance in both MNIST-M and Office-31 target domains.
- **Comprehensive Evaluation:** We evaluate our approach on multiple domain transfer tasks in Office-31 (e.g., Amazon → Webcam, Webcam → DSLR), demonstrating its adaptability to realistic scenarios.

This work contributes toward building domain-robust classifiers that can generalize well across varying environments, reducing the need for expensive labeled data in new domains and paving the way for more reliable deployment of AI systems in the wild.

2. Related Work

Domain adaptation has become an essential focus within transfer learning, especially for tasks where there is a lack of labeled data in the target domain. Domain adaptation has many applications in computer vision related works [6]. A variety of approaches have been suggested to tackle the issue of domain shift.

One of the early contributions in domain adaptation is the Office-31 dataset introduced by Saenko et al. [3], which has since become a standard benchmark for evaluating adaptation algorithms across visually diverse domains (Amazon, Webcam, DSLR). Conventional techniques, including Maximum Mean Discrepancy (MMD), have been extensively employed to minimize the distributional distance between source and target domains by aligning their feature distributions within a reproducing kernel Hilbert space.

In the domain of digit classification, MNIST → MNIST-M [1, 2] serves as a widely recognized synthetic benchmark utilized in the field of domain adaptation research. The shift in domain arises due to background noise and color in MNIST-M, making it ideal for evaluating the robustness of domain-invariant models.

Various recent studies have integrated CNNs with domain adaptation objectives to leverage deep hierarchical features. For instance, Sun et al. [3] demonstrated how deep CORAL can be integrated into CNN architectures for end-to-end adaptation. Other researchers have shown that data augmentation techniques can enhance domain robustness by increasing diversity in the source domain, especially when combined with domain alignment methods. A wide range of these approaches are used and algorithms are designed only for all datasets. For instance, Gopalan et al. [7] have proposed model-agnostic, open set domain adaptation approach using CNNs and recurrence plots for cross-domain classification.

Despite progress, many existing methods are computationally expensive or rely on adversarial objectives that are difficult to train. In contrast, this paper focuses on a simpler, effective, and interpretable framework using CNNs combined with CORAL and data augmentation, and validates its performance on both synthetic and real-world domain adaptation benchmarks. Deep learning (DL) falls within the domain of machine learning and is used in various fields [13].

Ajith et al. [4] demonstrated that non-adversarial, discrepancy-based domain adaptation methods—specifically combining MMD and CORAL losses with targeted feature transformations—can achieve accuracy levels comparable to complex GAN-based approaches. Their findings reinforce the effectiveness of hybrid loss strategies in learning domain-invariant representations, highlighting that carefully tuned, lightweight architectures can yield high generalization performance across domains without the computational overhead of adversarial training.

3. Materials and Methods

3.1. Datasets

To evaluate domain adaptation across both synthetic and real-world scenarios, we utilize three benchmark datasets that exhibit varying degrees of visual complexity and distributional shift.

Dataset	**Domain Description**	**Source**
MNIST	Handwritten digits	MNIST Database by Yann LeCun [1]
MNIST-M	Handwritten digits with textured backgrounds	Generated using the BSDS500 dataset [2]
Office-31	Real-world object images from different acquisition sources	Saenko et al. [3]

3.2. Data Preprocessing

- **Image resizing and normalization:** The MNIST images are maintained at their original resolution of 28×28 pixels and normalized to the $[0, 1]$ range. MNIST-M images, originally sized at 32×32 pixels, are resized to 28×28 to match the MNIST dimensions and are similarly normalized.
- **Channel adjustment:** MNIST, being grayscale with a single channel, is converted to a 3-channel RGB format by replicating the original channel, ensuring consistency with MNIST-M images.
- **Target-only normalization:** Since MNIST-M and Office-31 target domain datasets are unlabeled, only normalization is applied without augmentation, preserving their natural distribution.
- **Stratified sampling (Office-31):** Stratified sampling is used to maintain balanced class representation across domains like Amazon, DSLR, and Webcam. This helps reduce class imbalance and stabilize training.
- **Shuffling and batching:** Data loaders shuffle and batch both source and target domain images during training to ensure stochasticity and mitigate overfitting.

These preprocessing steps help standardize input formats, reduce domain gaps and maintain balanced representation across classes, thereby improving domain adaptation and feature alignment.

3.3. Data Augmentation

Data augmentation techniques are applied exclusively to the source domain to improve robustness and generalization capability.

- **Random rotation:** Digits and objects are randomly rotated within a specified angle range to simulate variations in orientation.

- **Horizontal and vertical flipping:** Source images are randomly flipped to introduce mirrored variations, especially useful in Office-31 domains like Amazon and Webcam.
- **Shifting and zooming:** Small random shifts and zooms are applied to MNIST digits and Office-31 images, mimicking different viewpoints and camera focus levels.
- **Shear transformations:** Affine shear transformations are applied to MNIST digits to emulate subtle distortions from handwriting or imaging artifacts.
- **Color jitter (Office-31 and MNIST):** Brightness, contrast and saturation are randomly altered in Office-31 and MNIST images to enhance resilience to lighting variations.
- **Random RGB Tinting:**Each color channel (R, G, B) is independently scaled by a random factor drawn from a uniform distribution over the range [0.5, 1.5], simulating variations in illumination and color intensity across channels.
- **Structured Background Blending (MNIST):** Original grayscale MNIST digits are converted to RGB and blended with a structured noisy background using a random binary mask and an adaptive alpha blending factor. This helps simulate real-world image noise and complex backgrounds.
- **Random Affine Transformations:** Digits are randomly rotated (up to $\pm 15°$) and translated (up to 10% in both axes) to improve spatial invariance and model robustness to alignment variation.
- **Custom Augmentation Pipeline:** All augmented images are resized to 28×28 and transformed using a composed PyTorch pipeline. Each image has an 80% chance of background blending and a 20% chance of only color and affine transformations.

By diversifying the source data, these augmentation strategies encourage the model to learn domain-invariant features, boosting performance on unseen target domains such as MNIST-M and Webcam.

4. Methodology

We assess our framework using two separate strategies: one employs a ResNet-50 backbone fine-tuned for cross-domain object classification on the **Office-31** dataset, while the other utilizes a lightweight CNN architecture to address digit recognition challenges posed by domain shift in the **MNIST to MNIST-M** setting.

4.1. CNN-Based Domain Adaptation using ResNet-50 (Office-31)

4.1.1. Feature Extractor

We employ a pre-trained **ResNet-50** [16] model as the base feature extractor. Initially trained on ImageNet, the model is fine-tuned using the source domain data from Office-31. The original classification head is removed and replaced with a fully connected layer corresponding to 31 object categories.

4.1.2. Optimization and Training

The model is trained using the **Adam optimizer** [17]. Batch normalization and dropout are incorporated to prevent overfitting. Training is conducted maximum for 30 epochs with a scheduled learning rate decay. This number is chosen based on prior empirical benchmark [15] and to ensure a fair comparison across methods. However we have employed other early stopping criteria based on validation loss. Data loaders ensure domain-specific sampling for each mini-batch.

4.1.3. Evaluation

Performance is primarily evaluated using quantitative metrics such as classification accuracy, precision, recall and F1-score on the target domain. To complement these results, we also generate qualitative visualizations, including t-SNE, per-class accuracy breakdowns, and confusion matrices, to better understand feature alignment and the model behavior across domains. This combination of quantitative and qualitative assessments provides a comprehensive evaluation of domain adaptation effectiveness.

4.2. Domain Adaptation with Custom Designed CNN on MNIST

4.2.1. Datasets

We utilized the standard MNIST dataset as the source domain, while MNIST-M serves as the target domain. MNIST-M is constructed by merging MNIST digits with RGB noise backgrounds, thus simulating a domain shift.

4.2.2. Data Augmentation Techniques

To improve generalization, MNIST training images were enhanced using random rotations (±15°), width and height shifts (±10%), shearing, zooming (0.8–1.2 range), and background coloring. These enhancements help to create a more robust feature representation to handle visual variability in MNIST-M.

4.2.3. Model Architecture

The proposed model comprises a compact CNN-based feature extractor and a separate classifier. The feature extractor includes three convolutional layers with 3×3 kernels and ReLU activations, each followed by batch normalization. Spatial downsampling is performed using 2×2 max pooling layers after the first two convolutional blocks. The output is then flattened and passed through a dense layer with 128 ReLU-activated units to generate feature embeddings. The classifier takes these 128-dimensional features as input and applies a dropout layer with a rate of 0.3 to prevent overfitting, followed by a dense output layer with 10 units and softmax activation for multi-class digit classification. The model is trained using the Adam optimizer (learning rate = 0.001) and categorical cross-entropy loss.

4.2.4. Evaluation

The evaluation primarily relies on standard quantitative measures, including accuracy, precision, recall and F1-score, computed on the target domain. To supplement these metrics, we employ qualitative visualizations like t-SNE and confusion matrices. The t-SNE plots provide a visual representation of how well the source and target features are aligned in the embedding space—tight, well-separated clusters often indicate higher classification performance. Similarly, the confusion matrix offers a direct view of class-wise prediction patterns, allowing us to estimate misclassification trends and intuitively assess overall accuracy. Together, these tools offer both a numerical and visual understanding of the model's effectiveness in domain adaptation.

5. Result and Analysis

5.1. Performance Evaluation Metrics

To assess the efficacy of the proposed domain adaptation models, standard classification metrics were employed. These metrics provide insights into overall accuracy and class-wise behavior, particularly in the presence of class imbalances or domain shifts. The primary metrics include:
Precision ($\frac{TP}{TP+FP}$) – proportion of correctly predicted positives out of all predicted positives, **Recall** ($\frac{TP}{TP+FN}$) – proportion of correctly predicted positives out of all actual positives, **F1-score** $\left(2 \cdot \frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}\right)$ – harmonic mean of precision and recall, **Accuracy** ($\frac{TP+TN}{TP+TN+FP+FN}$) – proportion of correctly classified samples overall, **Weighted Precision** $\left(\sum_{i=1}^{N} \frac{s_i}{S} \cdot \text{Precision}_i\right)$, **Weighted Recall** $\left(\sum_{i=1}^{N} \frac{s_i}{S} \cdot \text{Recall}_i\right)$, **Weighted F1-score** $\left(\sum_{i=1}^{N} \frac{s_i}{S} \cdot \text{F1}_i\right)$.
Where:

- TP, TN, FP, FN: True/False Positives and Negatives
- N: Total number of classes
- s_i: Support (number of true samples) for class i
- $S = \sum_{i=1}^{N} s_i$: Total number of samples
- Precision_i, Recall_i, F1_i: Metric values for class i

5.1.1. t-SNE Visualization

Interpretation: t-SNE visualizations help retain local neighborhood structure, allowing intuitive insight into how source and target features cluster. These plots are particularly useful for evaluating the degree of feature alignment achieved after applying domain adaptation methods like CORAL.

5.2. Analysis

Table 1: Baseline Performance of Traditional Machine Learning Models on Target Domain Evaluation

Model	Dataset	Accuracy (%)	W-Precision(%)	W-Recall(%)	W-F1-Score-ratio
SVM	A→D	4.02	5.16	4.54	0.0214
Naive Bayes	A→D	2.81	0.09	3.23	0.0020
Decision Tree	A→D	3.41	2.42	3.72	0.0187
Random Forest	A→D	2.61	2.13	4.57	0.0080
KNN	A→D	3.21	0.63	3.81	0.0097
SVM	MNIST→MNIST-M	0.28	0.75	0.29	0.0029
Random Forest	MNIST→MNIST-M	0.12	0.29	0.12	0.0007
Logistic Regression	MNIST→MNIST-M	0.15	0.63	0.15	0.0010

where:

- W-Precision, W-Recall and W-F1-Score-ratio refer to weighted precision, weighted recall, and weighted F1-score in range [0,1] respectively.
- Weighted metrics are used to reflect class imbalance in the target domain. Precision, recall and F1-score are weighted by class support.
- A→D represent that the training data is Office 31 amazon images and test data is office 31 DSLR images respectively.
- MNIST→MNIST-M represents the training data is MNIST and test data is MNIST-M .

From Table 1, we can observe that classical machine learning models such as SVM, Random Forest and Logistic Regression perform poorly in domain adaptation tasks like *A→D* and *MNIST→MNIST-M*. The models show very low accuracy, w-precision, w-recall and w-F1-scores. The reason is pretty simple, as these models operate on fixed input features and lack the flexibility to adapt to changes in data distribution. Even if alignment methods like CORAL or MMD are used externally, their effect remains limited because these models do not have a learnable feature transformation process. On the other hand, deep learning methods are capable of progressively adjusting feature representations and incorporating domain adaptation techniques directly into the training loop, which makes them more effective in dealing with domain shifts.

Table 2: Comparative Performance of Traditional and Deep Learning-based Domain Adaptation Techniques

Model	Dataset	Accuracy (%)	W-Precision(%)	W-Recall (%)	W-F1-Score-ratio
SVM	A→D	85.20	85.35	84.81	0.8327
NaiveBayes	A→D	85.33	88.42	88.79	0.8889
DecisionTree	A→D	62.85	63.08	64.85	0.6212
RandomForest	A→D	84.57	84.23	83.74	0.8333
KNN	A→D	87.48	88.61	85.21	0.8914
XGBoost (Autoencoder Features)	A→D	74.58	77.50	74.09	0.7197
CNN Basic model	MNIST→MNIST-M	37.37	59.00	37.00	0.3900
CNN + CORAL Loss	MNIST→MNIST-M	64.37	72.00	64.00	0.6600
CNN + MMD Loss	MNIST→MNIST-M	46.07	58.00	46.00	0.4700
Augmented CNN + CORAL Loss	MNIST→MNIST-M	61.19	70.00	61.00	0.6300
CNN + CORAL-SVM Loss	MNIST→MNIST-M	63.25	68.00	63.00	0.6400
CNN + CORAL Loss + MMD Loss	MNIST→MNIST-M	63.88	70.00	64.00	0.6500
CNN + CORAL-SVM Loss + MMD Loss	MNIST→MNIST-M	67.01	72.00	67.00	0.6800

- Refer to Table 1 for metric definitions and domain notations.

Table [2] shows a comparison of classical ML and deep learning models for domain adaptation across datasets like Amazon → DSLR and MNIST→MNIST-M. Among classical models, KNN achieved the highest w-F1-score (0.8914), showing strong performance in handling minor domain shifts via localized boundaries. Naive Bayes also performed well (w-F1: 0.8889), outperforming SVM (w-F1: 0.8327). Decision Tree performed poorly due to overfitting (w-F1: 0.6212), while Random Forest improved generalization (w-F1: 0.8333) due to ensembling. XGBoost with autoencoder features underperformed on Office-31 because the autoencoder was trained to minimize reconstruction error, not to make features domain-invariant or class-separable. As a result, its latent space still encoded domain-specific styles (e.g., lighting, backgrounds) instead of focusing on transferable semantic cues, and XGBoost overfit to these source-only patterns. Without explicit alignment (e.g., CORAL, DANN) or label-aware constraints, the learned features didn't generalize well to the shifted target domain. The enhanced performance of our method compared to the baseline arises from the integration of deep feature extraction and domain adaptation. Unlike the baseline, where conventional classifiers function on raw features, our methodology uses a pre-trained ResNet-50 to derive rich semantic representations. These deep features enhance classifiers such as SVM, XGBoost and Random Forests, providing them with superior discriminative power. To tackle domain shift, we utilize non-adversarial adaptation strategies like CORAL and MMD within the deep feature space. This guarantees that the learned representations are both resilient and domain-invariant, resulting in significantly improved cross-domain generalization. CORAL aligns the source and target domain features whereas MMD encourages alignment by minimizing this discrepancy during training. We evaluated CORAL-Enhanced CNN and Augmented CNN with CORAL Loss on the MNIST → MNIST-M domain adaptation task. The models, using CORrelation ALignment to align feature covariances, achieved w-F1-scores of 0.6600 and 0.6300 respectively, showing moderate adaptation success consistent with prior findings by Sun and Saenko [3]. Lastly, a deep CORAL-SVM with MMD yielded an w-F1-score of 0.6800, highlighting that strong feature representations can enable traditional classifiers to remain competitive in cross-domain scenarios.

Table 3: Accuracy comparison of different loss combinations for MNIST → MNIST-M domain adaptation in proposed model and in Ajith et al. [4].

Loss Function	**DDA**	**CORAL**	**MMD**	**DDA + CORAL**	**DDA + MMD**	**DDA + MMD + CORAL**	**MMD + CORAL**	**CORAL + SVM**	**CORAL + SVM + MMD**
Proposed Method	–	64.37	46.70	–	–	–	63.88	63.25	67.01
Ajith et al.[4]	70.00	66.50	72.50	77.16	81.67	78.50	–	–	–

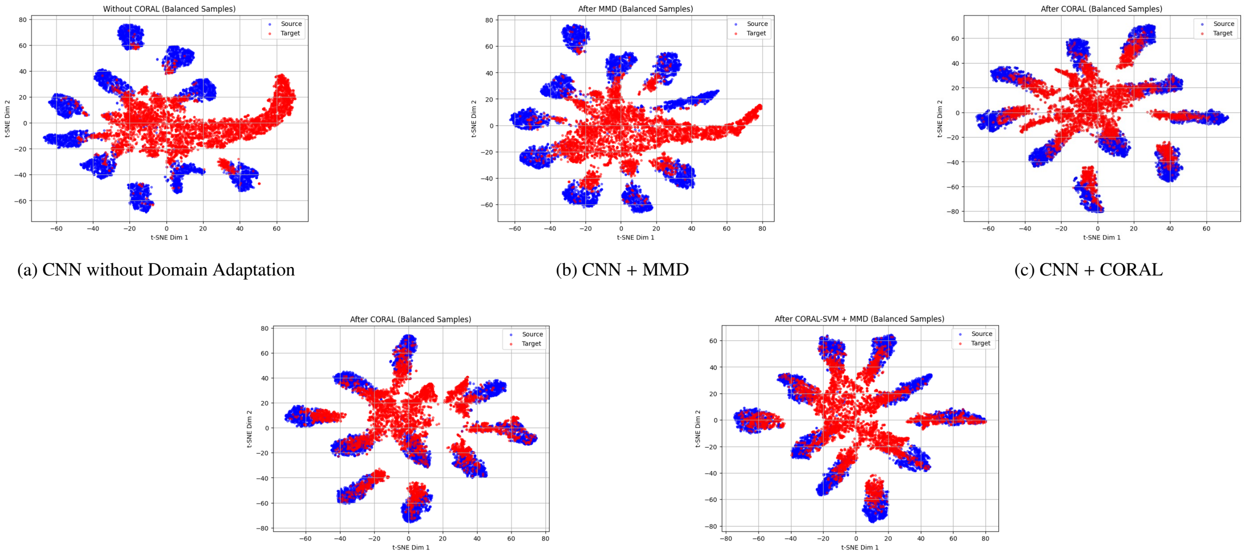
In our experiments, the MMD-based model underperformed significantly compared to CORAL-based models, achieving only 46.07% accuracy versus 64.37%. This gap is primarily due to our lightweight CNN, which produces shallow 128-dimensional features. MMD relies on kernel-based alignment in high-dimensional spaces and typically requires both expressive features and large batch sizes, conditions our model does not fully meet. Its class-agnostic nature also makes it less effective at preserving class boundaries in compact feature spaces. In contrast, CORAL aligns second-order statistics (covariances), which are more robust under such constraints and better suited to shallow models. By comparison, the improved MMD performance in Ajith et al. [4]'s work can be attributed to their deeper architecture and the DDA module, which likely produced more discriminative features. While MMD benefits from such complexity, our simpler model is better aligned with the stable and effective behavior of CORAL.

Where:

- **SVM** = A squared hinge-like loss applied to softmax probabilities, encouraging the correct class probability to exceed a margin.
- **DDA** = Discrepancy-based Domain Alignment, a learnable transformation module proposed by Ajith et al. [4].
- **CORAL** = Correlation Alignment, a loss function that aligns second-order statistics (covariances) of source and target features.
- **MMD** = Maximum Mean Discrepancy, a kernel-based loss that minimizes the distance between source and target feature distributions.

5.3. Understanding the Performance Gap of CNNs in Domain Adaptation

While CNN-based models consistently outperform classical machine learning approaches in domain adaptation tasks due to their capacity for learning deep, transferable features, our implementation intentionally adopts a lightweight CNN architecture. This choice reflects a trade-off between model complexity and practical efficiency, making it well suited for real-time or resource-limited settings. Although this architecture delivers noticeable improvements over traditional classifiers, especially when combined with CORAL loss for second-order statistical alignment, it may perform poorly compared to advanced adversarial models that explicitly enforce domain invariance. Methods like DANN, CoGAN and other GAN-based frameworks utilize adversarial objectives to suppress domain-specific cues, resulting in stronger generalization across large domain shifts. As demonstrated in the work of Ajith et al. [4], the integration of a deeper CNN architecture with a learnable DDA module led to improved domain alignment and classification accuracy. Their results highlight the benefits of using more expressive models and transformation-based alignment techniques, reinforcing the potential of scaling our approach through architectural enhancements or integration of similar alignment strategies.



(a) CNN without Domain Adaptation (b) CNN + MMD (c) CNN + CORAL

(d) CNN + CORAL + SVM (e) CNN + CORAL + SVM + MMD

Fig. 1: Comparison of t-SNE projections across different domain adaptation strategies on MNIST → MNIST-M

where :

Blue → source domain (MNIST)

Red → target domain (MNIST-M)

6. Loss Functions and Domain Alignment

To address the domain shift between the source and target datasets, we use a composite loss function that combines supervised classification with unsupervised domain adaptation. This allows the model to learn both accurate class boundaries and domain-invariant representations.

6.1. Cross-Entropy Loss

The primary objective is supervised classification in the source domain using categorical cross-entropy loss, which is defined as:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{1}$$

Here, y_i is the ground truth label and $\hat{y}_i$ is the predicted probability for the i^{th} sample.

6.2. CORAL Loss

To address domain shift across domains, we incorporated the *CORrelation ALignment* (CORAL) loss into our CNN-based architecture. CORAL specifically aligns the covariance matrices of the source and target feature representations. By minimizing the Frobenius norm of the difference between these covariance matrices, CORAL encourages the model to learn domain-invariant features. This helps the classifier trained on the labeled source domain generalize effectively to the unlabeled target domain.

In our implementation, the CORAL loss is optimized in conjunction with the standard classification loss through a weighted sum:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda \mathcal{L}_{\text{CORAL}}, \tag{2}$$

where $\mathcal{L}_{\text{cls}}$ denotes the cross-entropy classification loss on the labeled source domain, $\mathcal{L}_{\text{CORAL}}$ represents the CORAL alignment loss, and λ is a hyperparameter controlling the trade-off between classification and domain alignment.

Empirical results confirm that CORAL significantly improves performance in the target domain by enhancing domain alignment.

6.3. Maximum Mean Discrepancy (MMD)

Also, MMD is employed as a domain adaptation technique to reduce the distributional gap between both domains, which is critical for improving model generalization in cross-domain settings. Domain adaptation becomes necessary when a model trained on source dataset which is labeled performs poorly on target dataset (unlabeled) due to domain shift. MMD addresses this by aligning the feature distributions of both domains in a shared latent space.

By incorporating MMD as a domain alignment loss during training—alongside traditional classification loss, the model learns domain-invariant features that are both discriminative and transferable. The combined loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda \mathcal{L}_{\text{MMD}}, \tag{3}$$

where $\mathcal{L}_{\text{cls}}$ is the cross-entropy classification loss and $\mathcal{L}_{\text{MMD}}$ denotes the MMD alignment loss weighted by the factor λ.

6.4. CORAL′ with SVM-Inspired Margin

Alongside cross-entropy classification, we introduce a domain adaptation term $\mathcal{L}'_{\text{CORAL}}$ combining covariance alignment with an SVM-style margin penalty. The CORAL part aligns the second-order statistics of source and target features via the Frobenius norm of their covariance difference.

The margin term applies a squared hinge penalty to the source logits, encouraging the correct-class logit z_{i,y_i} to exceed a fixed margin m, thereby promoting confident predictions. This serves as an auxiliary regulariser within the alignment objective.

$$\mathcal{L}'_{\text{CORAL}} = \alpha \frac{\|C_S - C_T\|_F^2}{4d^2} + \beta \frac{1}{N} \sum_{i=1}^{N} \left[\max(0,\, m - z_{i,y_i}) \right]^2, \tag{4}$$

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{da}} \mathcal{L}'_{\text{CORAL}}, \tag{5}$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss, and λ_{da} controls the domain adaptation weight. Here, $m = 1$ was experimentally chosen after testing $m \in \{1, 0.9, 0.8, 0.7, 0.95\}$, as it provided the best balance between classification confidence and convergence stability.

6.5. Conclusion

The proposed domain adaptation framework demonstrates strong generalization performance across domains by effectively integrating advanced deep learning feature extractors, domain alignment techniques and robust classification strategies. Specifically, feature extraction using pre-trained deep convolutional networks such as ResNet-50 and EfficientNet variants provides highly transferable representations, minimizing the covariate shift between source and target domains [9]. Fine-tuning these networks on the source domain, followed by dimensionality reduction using PCA, helps retain discriminative components while reducing overfitting. Furthermore, domain alignment strategies such as CORAL [10] and Deep CORAL effectively minimize the domain discrepancy in feature space, thereby enhancing the model's ability to generalize to unseen target data. Incorporation of SMOTE [11] compensates for class imbalance by synthetically generating minority class samples, improving the recall and F1-score of underrepresented categories. Additionally, the integration of XGBoost classifiers over deep features boosts decision boundary sharpness, leading to superior precision and accuracy. These findings align with recent studies that emphasize hybrid adaptation frameworks for maximizing transfer accuracy [12]. This work also explored domain adaptation between the MNIST and MNIST-M datasets using a lightweight convolutional neural network trained with a hybrid loss function. Rather than relying on adversarial strategies like Domain-Adversarial Neural Networks (DANN), our approach incorporated *Correlation Alignment* (CORAL) to explicitly align second-order feature statistics between the source and target domains. Additionally, a squared hinge-like loss was incorporated to encourage the correct class probability to exceed a margin, complementing the CORAL alignment in maintaining discriminative features in the learned representation.

Empirical results show that the CORAL-SVM hybrid CNN framework achieved a test accuracy of 63.25% , while the CORAL-SVM + MMD hybrid CNN framework achieved a test accuracy of 67.01% on the MNIST-M dataset, significantly outperforming naive source-only classifiers. These results validate the effectiveness of combining CORAL-based domain alignment with a margin-inspired loss to improve cross-domain classification, suggesting that integrating margin-based objectives can enhance confidence and robustness of source predictions under domain shift.

Future work may explore the integration of gradient reversal layers, domain classifiers and attention-based feature extraction to further improve domain invariance by incorporating coral-svm loss. Also, stacking classifier can be constructed by selecting the top 5 performing first-stage models, where predictions are combined using a meta-model to enhance overall accuracy [14].

References

[1] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

[2] Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5), 898–916.

[3] Saenko, K., Kulis, B., Fritz, M., and Darrell, T. (2010). Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*.

[4] Ajith, A., Damaraju, G. S., & Gopalakrishna Pillai, G. (2024). Improved Discrepancy based Domain Adaptation Network using combined loss functions and Feature transformations. *Proceedings of the 2024 ACM Conference*, Association for Computing Machinery, New York, NY, USA. doi:10.1145/3627631.3627635

[5] V. Gaadha, A. S. Dev, V. Panicker, A. Sreerag, and A. Ranjit, "Enhancing CNN Performance Across Diverse Domains: Exploring the Efficacy of Domain Adversarial Neural Network with the Use of CNN for Improved Generalization in Deep Learning," in *Proceedings of the 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kamand, India, pp. 1–8, doi: 10.1109/ICCCNT61001.2024.10725470.

[6] Ajith, A., & Gopakumar, G. (2022). Domain Adaptation: A Survey. *Computer Vision and Machine Intelligence*, **586**, 591. doi:10.1007/978-981-19-7867-8_47

[7] Gopalan, S. K., Mohapatra, A. R., & Sujith, R. I. (2023). Cross-domain classification of dynamical states of nonlinear fluid dynamical systems using recurrence plots and convolution neural networks. *IEEE Transactions on Instrumentation and Measurement*, **72**, 1–9. doi:10.1109/TIM.2023.3326640

[8] B. Murugaraj and A. Joseph, "Performance Assessment Framework for Computational Models of Visual Attention," in *Proceedings*, pp. 345–355, Jan. 2018, doi: 10.1007/978-3-319-68385-0_29.

[9] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

[10] Sun, B., & Saenko, K. (2016). Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In *European Conference on Computer Vision (ECCV) Workshops*.

[11] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321–357.

[12] Wang, M., & Deng, W. (2018). Deep Visual Domain Adaptation: A Survey. *Neurocomputing*, 312, 135–153.

[13] Akshay, S., *et al.* (2024). Open Set Domain Adaptation for Classification of Dynamical States in Nonlinear Fluid Dynamical Systems. *IEEE Access*, **12**, 699–726. doi:10.1109/ACCESS.2023.3345456

[14] A. J. Tripathy, S. Suresh Rao, S. Ashwani, and N. P. T. V., "Integrating Explainable AI and Stacking Classifiers for Scalable Fraud Detection with PySpark," in *Proceedings of the 2025 3rd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, Coimbatore, India, pp. 1–6, doi: 10.1109/ICAECA63854.2025.11012160.

[15] Prechelt, L. (1998). Early Stopping – But When? In G. B. Orr & K.-R. Müller (Eds.), *Neural Networks: Tricks of the Trade* (pp. 55–69). Springer, Berlin, Heidelberg.

[16] B. Koonce, "ResNet 50," in *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization*, Berkeley, CA: Apress, 2021, pp. 63–72, doi: 10.1007/978-1-4842-6168-2_6.

[17] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2017. [Online]. Available: `https://arxiv.org/abs/1412.6980`