Intelligent Automation & Soft Computing

Tech Science Press

DOI: 10.32604/iasc.2024.055074

ARTICLE

Robot Vision over CosGANs to Enhance Performance with Source-Free Domain Adaptation Using Advanced Loss Function

Laviza Falak Naz[1], Rohail Qamar[2,*], Raheela Asif[1], Muhammad Imran[2] and Saad Ahmed[3]

[1]Department of Software Engineering, NED University of Engineering & Technology, Karachi, 75270, Pakistan

[2]Department of Computer Science & Information Technology, NED University of Engineering & Technology, Karachi, 75270, Pakistan

[3]Department of Computer Science, IQRA University, Karachi, 75500, Pakistan

*Corresponding Author: Rohail Qamar. Email: rohailqamar@cloud.neduet.edu.pk

Received: 15 June 2024 Accepted: 05 August 2024 Published: 31 October 2024

ABSTRACT

Domain shift is when the data used in training does not match the ones it will be applied to later on under similar conditions. Domain shift will reduce accuracy in results. To prevent this, domain adaptation is done, which adapts the pre-trained model to the target domain. In real scenarios, the availability of labels for target data is rare thus resulting in unsupervised domain adaptation. Herein, we propose an innovative approach where source-free domain adaptation models and Generative Adversarial Networks (GANs) are integrated to improve the performance of computer vision or robotic vision-based systems in our study. Cosine Generative Adversarial Network (CosGAN) is developed as a GAN that uses cosine embedding loss to handle issues associated with unsupervised source-relax domain adaptations. For less complex architecture, the CosGAN training process has two steps that produce results almost comparable to other state-of-the-art techniques. The efficiency of CosGAN was compared by conducting experiments using benchmarked datasets. The approach was evaluated on different datasets and experimental results show superiority over existing state-of-the-art methods in terms of accuracy as well as generalization ability. This technique has numerous applications including wheeled robots, autonomous vehicles, warehouse automation, and all image-processing-based automation tasks so it can reshape the field of robotic vision with its ability to make robots adapt to new tasks and environments efficiently without requiring additional labeled data. It lays the groundwork for future expansions in robotic vision and applications. Although GAN provides a variety of outstanding features, it also increases the risk of instability and over-fitting of the training data thus making the data difficult to converge.

KEYWORDS

Cosine generative adversarial network; cosine embedding loss; generative adversarial networks; source free domain adaptation; unsupervised learning; hyper-parameter

1 Introduction

In Convolutional Neural Networks (CNNs), a common method is supervised learning which necessitates using labeled source data to train on similar targeted data for predicting the outcome.

However, in typical scenarios involving large-scale real-world applications, one often has access to extensive but not annotated data leading to unsupervised learning getting explored. However, unsupervised learning is still an active area of field while domain adaptation appears as a strategy for rapid training to its intended domains. One of these challenges is called the domain shift [1–4] and it entails training a model on source data so that it performs well on target data with slightly varying characteristics. Some scholars have addressed this problem by using techniques like labeled source/target data or labeled source/unlabeled target data [5–8]. In practice, many realistic cases involve applying Unsupervised Domain Adaptation (UDA) to minimize distribution mismatches using unlabeled target instances [9]. Nevertheless, earlier UDA work usually utilizes both source and target datasets hence requiring us to adapt from the source domain. In this instance, however, we would have to resort to unsupervised source-free domain adaptation since labeled source data may not be available post-deployment due to privacy issues or computational constraints associated with sizeable datasets [10]. Generative Adversarial Networks (GANs) encompass a generator and discriminator in our research study. GANs constitute a two-player game that trains two neural networks: the generator network and the discriminator network. They are aligned in opposition in a zero-sum game. The job of the generator is to generate realistic data instances, while the job of the discriminator is to distinguish between real and generated data. Optimization on training continues when the generator becomes better at its output to fool the discriminator, and the discriminator becomes better at detecting fake data [11–13]. This process involves the generator synthesizing outputs during GAN training as shown in Fig. 1 whereas the discriminator assesses their authenticity and classifies them as "Real" or "Fake" ones [14].

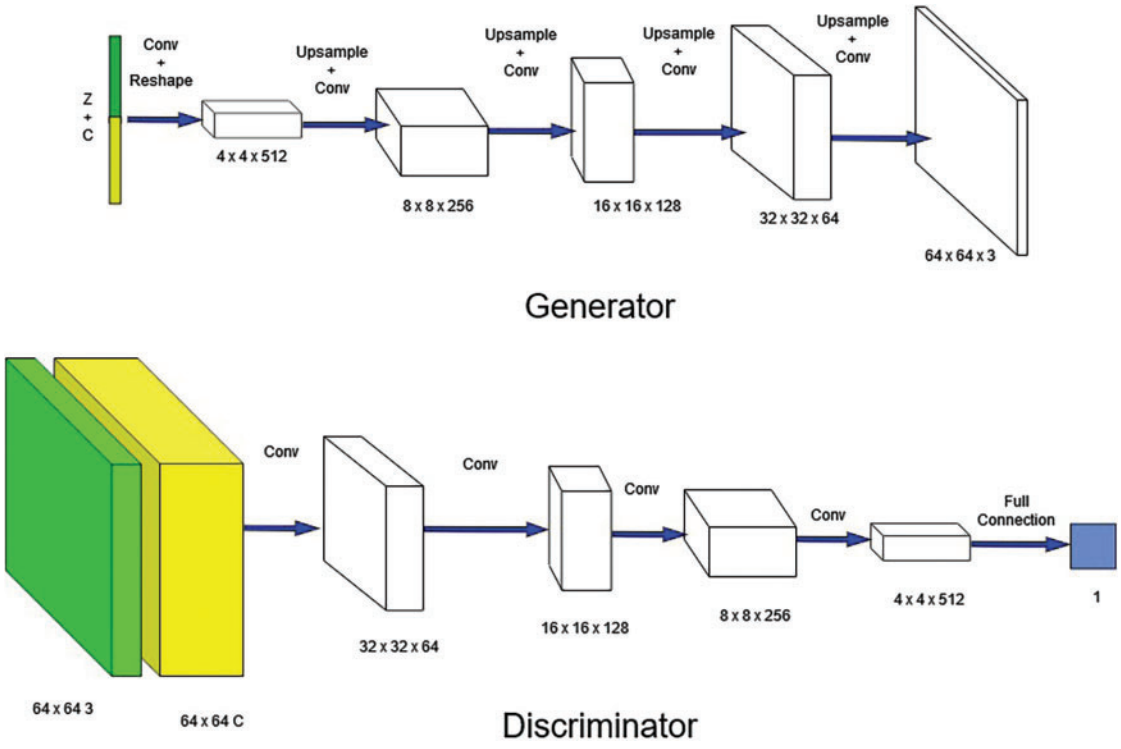

Figure 1: GAN architecture

The outcome of the generator is determined by the extent to which it misleads the discriminator. Successful fooling of the discriminator attracts rewards while unsuccessful attempts carry a punishment. Calculations of both generator and discriminator are described by Eqs. (1) and 2. To address this, our study proposes a GAN-based framework for unsupervised source-free domain adaptation that promotes model generalization across domains.

$$\theta_g \frac{1}{m} \sum_{i=1}^{m} \left[logD\left(x^i\right) + log\left(1 - D\left(G\left(z^i\right)\right)\right)\right] \tag{1}$$

$$\theta_g \frac{1}{m} \sum_{i=1}^{m} log(1 - D\left(G\left(z^i\right)\right)) \tag{2}$$

Instead, our proposed approach does not need any source data or labeled target data. It rather requires for pre-trained model. In our methodology, we provide the encoder of GAN's generator with pre-trained models. The other two parts of the GAN, i.e., the decoder and discriminator are randomly initialized in our methodology. We discovered that random initialization of decoder and discriminator components can result in generating any outputs leading to corruption of the pre-trained encoder state. So to avoid it, we freeze out the encoders' encoding layers so that their initial conditions remain unaffected by the decoder's or discriminators' effect on them in two parts processes of GANs. When enough training has been done on the decoder and discriminator, we can unfreeze these encoding layers through such a strategic progression where we preserve integrity in what was trained about their architecture as per what was provided before.

Our approach is designed for Source-Free Unsupervised Domain Adaptation (SF-UDA)–thus it offers a robust framework. Additionally, among others, we propose using cosine embedding loss for improved feature extraction from the target dataset. To boost performance and stabilize the GAN training process, advanced techniques are embedded within it. The use of the Two Time-Scale Update Rule (TTUR) together with Soft Label Smoothing inside D offers significant improvement in the stability and quality of GAN results. TTUR allows for modifying learning rates for both generator and discriminator networks during GAN training optimization. This optimization technique enhances GAN stability and quality, especially over longer training runs. Instead of hard (real/fake) decision boundaries are used by soft label smoothing is when answering whether a sample is genuine or bogus as provided to D. Consequently, this method increases the certainty level of authenticity determination by D resulting in richer gradients driving betterer signal accumulation M towards stable behavior in practice Finally, we noted that making more complex both generator's and discriminator's networks by adding more layers significantly improves GAN's performance. The use of deep convolutional GANs with multiple layers increases image quality, hence making the algorithm more effective.

Here is how the paper is organized. Section 2 reviews some related work in literature. The methodology and training process of our proposed model is mentioned in Section 3 while Section 4 specifies the results of our proposed approach. The research limitations are discussed in Section 5. Furthermore, the future directions are explained in Section 6. Finally, the research draws its conclusions in Section 7.

2 Literature Review

Domain adaptation is an area of much interest in current research [15,16], because it has implications in practice. It involves training models in one domain while the expectation is that they should perform well in another domain. A comprehensive review of domain adaptation was presented in 17, which highlighted the limitations of traditional machine learning when the source and target

datasets belong to the same domain. Such a scenario is rare in the real world. This review introduced domain adaptation and categorized it based on category gaps. An innovative approach proposed earlier transforms domain adaptation into supervised learning. The method augments both source and target datasets, and then trains algorithms for supervised learning 18. This strategy, known as augmentation, uses a substantial number of annotated source data and labeled target data to enhance performance [17]. Object recognition is the most significant step in enabling home robots to work with the ability of humans. This becomes more important when we consider tasks that involve recognizing known objects, even while surrounded by unknown ones. In this paper, we present the Continual Open Set Domain Adaptation for Home Robot (COSDA-HR) dataset, which jointly tackles the challenges presented by domain adaptation, open-set recognition, and continual learning. The dataset targets naturally arranged objects in a room. Objects are trained while being held, hence working towards enhancing teaching systems for home robots. The authors stress the need for approaches to handle multiple challenges together, noting the limitations of current methods [18]. However, problems arise from differences between source and target data sets as a result of obtaining labeled target data. Here, unsupervised domain adaptation helps by training models with labeled data to predict unlabeled target data [19–21]. Recently, Pedro O. Pinheiro developed a method that uses similarity-based classifiers. In this technique, the model learns from each source category and prototype vectors and then classifies target domain images based on these prototypes [22].

Semantic segmentation which is very demanding as far as information processing is concerned has experienced extensive work on domain adaptation [23–27]. The study conducted in [28] on unsupervised domain adaptation for semantic segmentation identifies areas for further research that can address its limitations. The Unsupervised Domain Adaptation method was proposed in [29], adopting a cycle consistency framework that helps to align source/target data leading to better results than otherwise would be possible. In cases where there are few labeled targets available, we need source-free DA methods indeed. The author [30] addressed this issue through self-learning, active learning, and use of augmented training sets (such as GAN). Also adapted are pre-trained models for related target domains given domain adaptation, or transfer learning. However, problems like noisy labels and limited source data require unique solutions. Consistency-Based Semi-Supervised Domain Adaptation (CB-SSDA) is one of the most recent techniques in this regard [31]. With CB-SSDA, consistency-based regularization helps to generate stable predictions over different inputs using labeled and unlabeled target domain data. This approach has shown remarkable efficacy on benchmark datasets like Office [31] and Office-Home [32]. Our study draws upon transfer learning, adaptive pre-training, batch normalization (BN) layers, multi-level domain invariant features, and advanced knowledge distillation (KD) schemas [33]. The prerequisite for these methods is that there should be a pre-trained classifier that can help bridge the resource gap between high-resource and low-resource domains [34]. Classification accuracy is improved through indirect learning involving inverse supervision [35]. The method also reduces the impact of noisy labels minimizing incorrect feedback probability as stated earlier [36].

A crucial and integral aspect of AI involves the use of ensemble methods such as Negative Ensemble Learning (NEL) which combines models to give a better overall performance [37]. Adaptive pseudo-label refinement (AdaPLR) is an adaptive pseudo-label refinement method that handles domain shift problems by improving pseudo-label quality through adaptive noise filtering [38]. AdaPLR makes use of Disjoint Residual Labels to improve ensemble diversity [39]. Domain adaptation has also penetrated medical research [40]. The authors in [41] were concerned with domain shifts in medical image segmentation. They put forward source-free domain adaptation which minimizes the entropy lost at the target domain, leading to excellent results [42]. Studies on GAN-based methods have

been done with or without source data and have yielded some good results in [42–50]. By unifying discriminative modeling, untied weight sharing, and GAN loss under the ADDA framework, this paper proposes an improved technique than others currently available in literature [51]. Real-world applications require this important field of study. How do CB-SSDA, SFDA, ensemble methods, and AdaPLR address these challenges while raising performance levels across diverse domains? These factors are driving innovation towards domain adaptation as a powerful strategy in different applications including GAN optimization, indirect learning or even source-free domain adaptation today because of all its capabilities.

In unsupervised domain adaptation, the AdaPLR method is presented as a unified approach to adaptive noise filtering and progressive pseudo-label refinement [52]. Using the source model, target pseudo-labels are created, and a set of target samples with various stochastic input augmentations and feedback mechanisms are used to train each ensemble member. The approach uses Negative Ensemble Learning (NEL) loss and Disjoint Residual Labels to make the ensemble more diverse and cut down on noise in the pseudo-labels. The proposed bound-together procedure for AdaPLR is portrayed in Fig. 2.

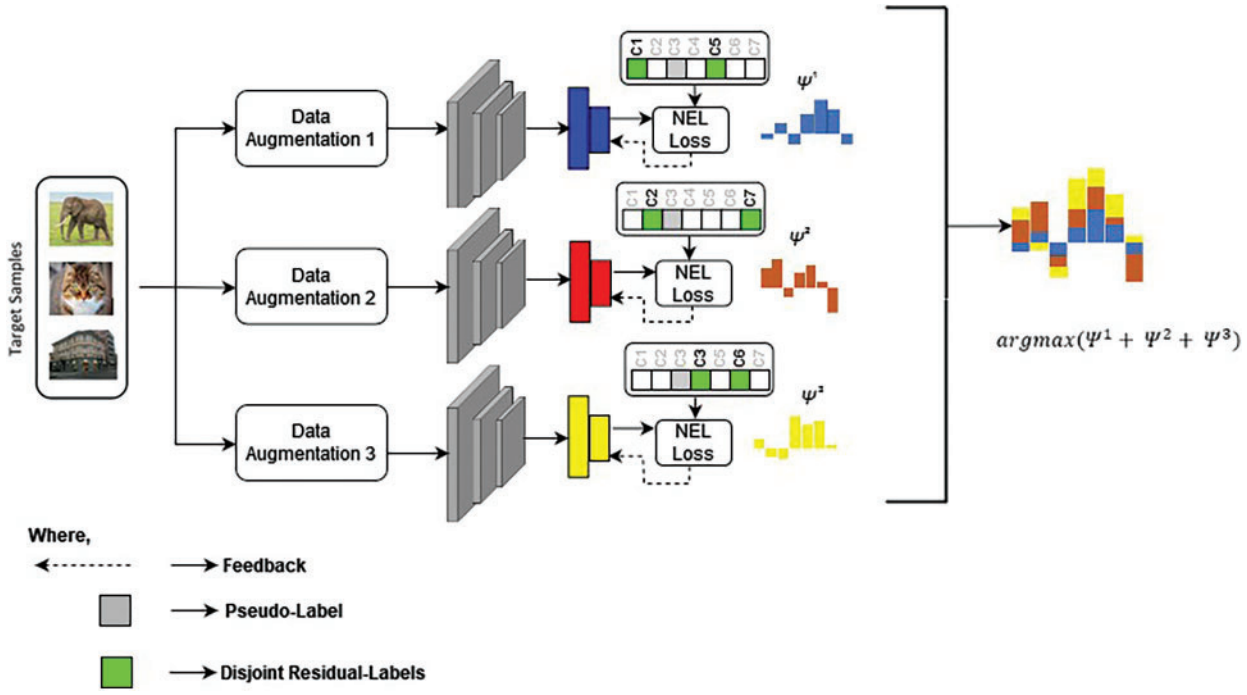

Figure 2: AdaPLR architecture [52]

AdaPLR is a method that uses Negative Ensemble Learning and Disjoint Residual Labels to deal with noise in unsupervised domain adaptation. With the inferred target pseudo-labels, the method trains each ensemble member using various stochastic input augmentation and feedback mechanisms. The argmax function is used to combine the predicted probabilities of each ensemble member to determine the maximum probability. In comparison to other current, cutting-edge unsupervised domain adaptation methods and techniques, AdaPLR has performed better in numerous domain adaptation benchmark tests.

A great change of thought has occurred in human-robot interaction because of the combination of large-scale language models and robotic systems that have unmatched natural language understanding and task execution capabilities. In this review paper, we focus on recent developments in LLMS with respect to improving their structures and performances, especially for multimodal input handling, high-level reasoning, and plan generation. Additionally, it examines how existing methods such as LLMs can be used to accomplish complex tasks in robotics from classical probabilistic models to metric-based approaches or value functions that enable optimal decisions. However, these advancements raise significant issues in the field such as ethical considerations, context awareness, and data privacy matters among others. This present work is the first study that comprehensively explores developments and concerns of LLMs in Human-Robot Interaction (HRI) based on current progress [53].

3 Methodology

In this section, we present a detailed methodology for implementing Robot Vision over COS-GANs with improved performance using source-free domain adaptation models and cosine embedding loss. Domain adaptation is the solution for a model that has been trained in one domain but has to perform well in another. A critical issue addresses domain shift in which the differences between the source and target domains degrade model accuracy. Traditional models in machine learning assume training and testing data are drawn from an identical distribution, which is rarely the case in real-world applications. The domain adaptation techniques go a step beyond this: their goal is to adapt a model to a target domain, often without using any data labeled in this domain. The common approaches to domain adaptation are: (i) Supervised Domain Adaptation: Here the adaptation is with labeled data in both source and target domains. This is the least practical method since it is very hard to get labeled data in the target domain. (ii) Unsupervised Domain Adaptation (UDA): A scenario with labeled source data along with unlabeled target data is thereby more applicable to the realistic scenario where labeling target data is expensive or very unrealistic. (iii) Source-Free Domain Adaptation: It operates on an already pre-trained source model and the unlabeled target data in a way that helps solve privacy issues and computational constraints in deploying large source datasets.

3.1 Source-Free Domain Adaptation

3.1.1 Definition and Significance

Source-free domain adaptation (SFDA) is a technique in machine learning where the adaptation of a pre-trained model to a new target domain is performed without access to the source domain data. This approach is particularly significant in scenarios where the source data cannot be used due to privacy concerns, data size constraints, or other practical limitations.

3.1.2 Importance in Robotic Vision

By leveraging SFDA, robotic vision systems can achieve better generalization and adaptability, enhancing their performance and reliability in various applications. CosGAN, with its use of cosine embedding loss and GAN-based architecture, exemplifies a robust framework for SFDA, promoting model generalization across domains and improving the overall performance of robotic vision systems.

3.2 Problem Formulation

Our aim is to generate a model for domain adaptation that works between a source dataset Xsource and target dataset Xtarget by improving task performance on the latter while transferring knowledge from the former most effectively. In this paper, we are particularly interested in image classification where each image is labeled y.

3.3 Architecture Overview

We have two main components: the Generator Network (G) and the Discriminator Network (D). The former transforms source domain images into target domain style while the latter differentiates between generated images and target domain ones.

3.4 Training Objective

This study proposes a GAN-based approach called CosGAN which incorporates cosine embedding loss for performing the task of unsupervised domain adaptation without source data. Fig. 3 shows the proposed CosGAN architecture. In SF-UDA, we have access neither to source images XS nor to actual labels of target images XT. Listed below are the several benefits of the proposed CosGAN model over traditional loss functions used in similar GAN architectures:

1. **Cosine Embedding Loss:** This loss helps in better feature alignment between source and target domains, improving the performance of unsupervised domain adaptation.
2. **Two Time-Scale Update Rule (TTUR):** This technique stabilizes the GAN training process by adjusting learning rates for the generator and discriminator separately, leading to more stable and higher-quality outputs.
3. **Soft Label Smoothing:** This method helps to prevent the discriminator from becoming too confident in distinguishing between real and fake samples, thus improving the overall stability and performance of the GAN.

3.4.1 Model Working

This figures demonstrates a composite architecture of the CosGAN implementation and it's working through-out the model. The first step includes the augmentation of Target Image XT to an Augmented Target Image. This XAT serves as the input for the Generator Module, L1 Loss Calculation, and the Discriminator. The Generator is intended to generate the Image XG while working with the weights. This Generator comprises an Encoder and a Decoder which work in parallel for the generation of XG. The assumptions underlying the generative model include: (i) A pre-trained model is available: The assumption behind the generative model is the availability of a pre-trained model; hence, it performs the best for the encoder part of the GAN. (ii) Success for cosine embedding loss: It assumes that cosine embedding loss will be effective in aligning features across two domains, one from the source and another from the target. (iii) Training Stability through TTUR: The Two Time-Scale Update Rule will hopefully keep the training process stable, even in unsupervised and source-free scenarios. (iv) Domain adaLabel Distribution: Soft label smoothing is expected to provide a better performance discriminator by alleviating overfitting to the fake data produced during training.

The experimental setup for evaluating CosGAN includes several key components: (i) Dataset Selection Criteria: The datasets selected for evaluation are benchmarked datasets such as PACS, which provide a diverse set of images across different domains, including Art, Cartoon, and Sketch. (ii) Preprocessing Steps: The preprocessing steps involve standard image preprocessing techniques, including normalization, resizing, and augmentation to ensure the model can generalize well across

different domains. (iii) Hyperparameter Tuning: The learning rate, batch size, and weight decay are optimized with rigorous hyperparameter tuning. A grid search was conducted to select these hyperparameters. (iv) Validation Techniques: These are procedures for the validation, for example, dividing the dataset between training and validation datasets, with some cross-validation methods to ensure the output is robust. The developed evaluation of the model will encompass measures like performance, classification accuracy, clustering accuracy, and GAN loss, among other loss parameters, such as cosine embedding loss. (v) Evaluation Metrics: CosGAN is evaluated in terms of classification accuracy (ACC), Adjusted Rand Index (ARI), and loss parameters, which are GAN loss, cosine embedding loss, and advanced loss function, among others.

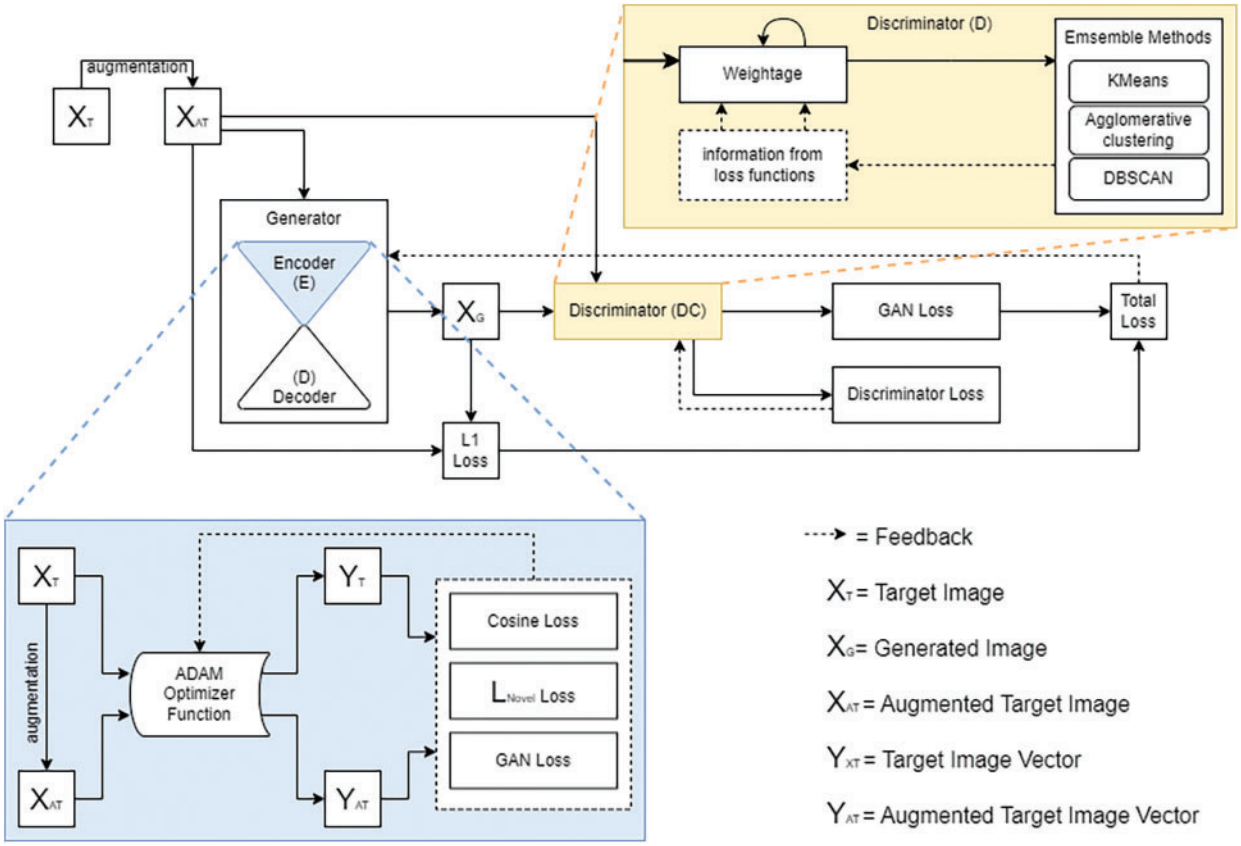

Figure 3: Overview of proposed CosGAN architecture

3.4.2 Working of Encoder

The Encoder works for the optimization of the *XT* and *XAT* to yield *YT* and *YAT* which determine the applicable Cosine, *LNovel*, and GAN losses. These Loss Parameters serve as a significant parameter to identify classification weights for the Ensemble methods in a later phase.

3.4.3 Working of Discriminator

The discriminator collects feedback from the loss functions and also calculates the weightage after applying the Ensembled models. This Weightage is recomputed over EPOCHs time iterations over

the applied Batch Size of 64 for a better compute time and a suitably tuned hyperparamater which contributes significantly towards the overall accuracy of the methods.

For the purpose of adapting the source model onto the target data, the fully connected layer of the trained source model MS is discarded. Now, we use this MS as an encoder in GAN whereas a Decoder D along with a Discriminator DC is initialized randomly. The standard training procedure of GAN with a pre-trained encoder, a randomized decoder, and a randomized discriminator would cause no good to the pre-trained encoder. Instead, due to the randomly generated outputs, the encoder will be disturbed negatively. In this proposed architecture, we freeze the encoding layers to prevent the pre-trained state of the encoder until the decoder and discriminator are trained to a fair point. At this point, we unfreeze the encoding layers to update their weight.

Our overall training objective is to optimize the Generator network G and Discriminator network D in a GAN-like fashion while incorporating the cosine embedding loss:

$$\min_{G}\max_{D} \boldsymbol{G}(G, D) = \mathbb{E}xtarget \sim X_{target}\left[\log D\left(X_{target}\right)\right] \tag{3}$$

$$\mathbb{E}xtarget \sim X_{source}\left[\log\left(1 - D\left(G\left(X_{source}\right)\right)\right)\right] \tag{4}$$

$$\lambda_{cosine}\mathbb{E}\left(X_{source}, X_{target}\right) \sim X_{source}xX_{target}\left[L_{cosine}\left(X_{source}, X_{target}\right)\right] \tag{5}$$

where λ cosine controls the influence of the cosine embedding loss.

3.5 Algorithmic Steps

Our training process alternates updates of the Generator network with those of the Discriminator network. Each iteration seeks to minimize overall loss from the generator, and maximize it from the discriminator:

1. Sample batch of target domain images X_{target}.
2. Update Discriminator D using real target images & generated images.
3. Sample batch of source domain images X_{source}.
4. Update Generator G using GAN loss & cosine embedding loss.

3.6 Evaluation Metrics

We use multiple evaluation metrics to ensure a comprehensive assessment of our proposed methodology.

3.6.1 Classification Accuracy

Classification accuracy (ACC) is one of the most fundamental metrics that measure how many instances were correctly classified out of all tested instances in the targeted test data set after adaptation has taken place. It can be calculated as follows:

$$Acc = \frac{N_{correct}}{N_{total}} \tag{6}$$

where $N_{correct}$ is the number of correctly classified instances and is the total number of instances.

3.6.2 Clustering Accuracy

Clustering efficiency we measure by Adjusted Rand Index (ARI), which quantifies the similarity between ground truth class labels and predicted cluster assignments considering all pairs for instance comparing their relationship with each other:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] \Big/ \binom{N}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] \Big/ \binom{N}{2}} \tag{7}$$

where n_{ij} is the number of instances that are in the same class in the ground truth and are assigned to the same cluster in the prediction. a_i is the number of instances in the same class as instance i in the ground truth, and b_j is the number of instances in the same cluster as instance j in the prediction.

3.6.3 Loss Parameters

To assess how much better our advanced loss function performs, we calculate losses (L) on both the source and target domain datasets. GAN loss, Cosine Embedding Loss, and Advanced Loss Function are the loss parameters formulated as per the equation:

$$L_{GAN} = \mathbb{E}_{x_{targert}} \sim X_{target}\left[\log D\left(X_{target}\right)\right] + \mathbb{E}_{x_{source}} \sim X_{source}\left[\log\left(1 - D(G(X_{source}))\right)\right] \tag{8}$$

$$L_{cosine} = \lambda_{cosine}\mathbb{E}_{\left(x_{source}, x_{target}\right)} \sim X_{source} \times X_{target}[L_{cosine}(x_{source}, x_{target})] \tag{9}$$

$$L_{Advanced} = \lambda_{advance}\mathbb{E}_{\left(x_{source}, x_{target}\right)} \sim X_{source} \times X_{target}\left[advanced\left(x_{source}, x_{target}\right)\right] \tag{10}$$

where λ_{cosine} and $\lambda_{\text{advanced}}$ are weighting factors for the cosine embedding loss and advanced loss function, respectively. The accuracy of classification (ACC) reflects the performance of the model in the target domain, where a higher ACC indicates successful adaptation and generalization. The Adjusted Rand Index (ARI) measures clustering quality. It becomes larger when predicted clusters are better aligned with true class labels, i.e., more suitable for feature representation learning. Analyzing the loss parameters (L_{GAN}, L_{Cosine}, and L_{Advanced}) sheds light on the contribution of individual loss components. These loss values reduction shows the efficiency of corresponding loss functions in guiding the optimization process.

3.7 Hyperparameters Tuning

In order to achieve optimal results during our study, an extensive search was carried out over hyperparameters such as learning rate (η), batch size (B), and weight decay (λ). The schedule for the learning rate follows Eqs. (2) and (3) incorporates weight decay into the training objective.

The choice of optimal hyperparameters is crucial for adjusting the behavior and convergence of the model. We investigated systematically how different hyperparameters impact the performance of the model such as learning rate (η), batch size (B), and weight decay (λ).

Learning rate is one of the most important hyperparameters that controls step size during gradient descent optimization. We adopted a grid search strategy to tune it so that we can find out what value makes it possible to converge quickly at optimal solutions. In other words, the learning rate (η) is updated based on epoch t iteratively by the following schedule:

$$\eta\,(t) = \frac{\eta_{initial}}{(1 + \alpha t)^{\gamma}} \tag{11}$$

where η_{initial} is the initial learning rate, α controls the learning rate decay rate, and γ modulates the decay's steepness. Similarly, batch size (B) was optimized through an extensive search across a range of values because it directly affects noise in gradient updates and the convergence speed of the model. Finally, weight decay (λ) was incorporated as a regularization term to avoid overfitting. Loss function with weight decay is given by:

$$L_{final} = L_{GAN} + \lambda_{cosine} L_{cosine} + \lambda_{L1} \left|\left| G\left(X_{source}\right) - X_{target} \right|\right| + \lambda \sum_{i} ||\theta_i||_2^2 \tag{12}$$

where θ_i represents the parameters of the model.

3.8 Cosine Embedding Loss

Cosine embedding loss is a metric learning loss by cosine similarity between the feature representations in source and target domains; this ensures the similarity of similar instances from either domain and, in turn, helps better domain adaptation. In contrast, the latter will help enforce some kind of consistency on the features across domains so that the distance of angles between embeddings of the same class is reduced. Cosine embedding loss has been proposed in order to improve alignment between source and target domains feature representations. Given a pair of instances ($x$source, $x$target), our objective is to minimize the cosine distance between their feature embeddings:

$$L_{Cosine}\left(X_{source}, X_{target}\right) = \frac{1}{2}\left(1 - \frac{\left\langle G\left(X_{source}\right), G\left(X_{target}\right)\right\rangle}{||G\left(X_{source}\right)|| \left|\left| G\left(X_{target}\right)\right|\right|}\right) \tag{13}$$

where (x_{source}) and (x_{target}) represent the feature embeddings obtained from the Generator network for the source and target domain instances, respectively. This loss term enforces similarity between domain-specific features.

3.9 Ensemble Methods

We use ensemble methods such as KMeans, Agglomerative Clustering, and DBSCAN to refine clustering results by combining different method outputs through majority voting that assigns instances into clusters which provides diverse perspectives on instance grouping hence reducing chances of selecting suboptimal clusters. We combined the individual decisions of these ensemble methods using a majority voting scheme. Given an instance x, the ensemble output (x) is determined as:

$$E\left(x\right) = argmax_c \sum_{i=1}^{M} \left[c = C_i\left(x\right)\right] \tag{14}$$

where M represents the number of ensemble methods and (x) denotes the cluster assigned to x by the ith ensemble method. Our experiments indicate significant improvement in clustering accuracy brought about by these ensemble methods which effectively capture complex underlying patterns within data distribution. Therefore, through systematic tuning of hyperparameters coupled with the integration of ensemble methods, we were able to greatly boost both model robustness and overall performance as shown by empirical results presented in Section 4.

3.10 Advanced Loss Function

In order to improve our methodology further, we propose an advanced loss function that combines GAN-related as well as cosine embedding components. This new loss function is based on the idea of encouraging better alignment between generated target domain samples and actual ones. We achieve this by incorporating the GAN loss, the cosine embedding loss, and a novel term that encourages closer feature representations of generated and target domain samples. The advanced loss function L_{advanced} is formulated as follows:

$$L_A\left(X_s, X_T\right) = L_{GAN}\left(X_s, X_T\right)\lambda_{cosine} \cdot L_{cosine}\left(X_s, X_T\right)\lambda_{novel} \cdot L_{novel}\left(X_s, X_T\right) \tag{15}$$

where L_{GAN} and L_{cosine} represent the GAN loss and the cosine embedding loss between the source and target domain samples, respectively. The new term L_{novel} is designed to encourage similarity between the feature vectors extracted from the generated source domain sample and the target domain sample. This term is defined as:

$$L_{novel}\left(X_s, X_T\right) = ||\phi\left(G\left(X_s\right)\right) - \phi_T(X_T)\,||_2^2 \tag{16}$$

where ϕ_S and ϕ_T are feature extraction functions for the source and target domain samples, respectively. (x_S) is the generated target domain sample from the source domain sample x_S.

The weighting factors λ_{cosine} and λ_{novel} control the relative influence of the cosine embedding loss and the novel term in the overall loss. Again in order to improve performance of our methodology, we propose an advanced loss function that leverages both GAN-related and cosine embedding components. This novel loss function is derived based on the idea of promoting better alignment between the generated target domain samples and the actual target domain samples.

3.11 Implementing and Training

We used the PyTorch framework for implementing our methodology where Generator and Discriminator networks are built using convolutional architectures. Network parameter updates were done using Adam optimizer while training was performed on the source domain dataset before adapting it to target domain through multiple epochs. During training process, network parameters get updated via backpropagation with respect to defined loss functions.

4 Experimental Results

This section presents detailed experimental results and analysis which demonstrate effectiveness of our proposed methodology. For instance, Table 1 provides ResNet18 classification accuracy on PACS with AdaPLR (a deep learning algorithm for image classification). Furthermore, Table 2 shows results obtained from PACS by comparing CosGAN against single source UDA without source data on NEL [33]. TSNE [50] visualization in Fig. 4 compares ground truth, pseudo labels from the pre-trained source model, and the target labels from the encoder trained with CosGAN by using K-Means clustering. The results show a sharp increase in average from 72.4 percent in AdaPLR to 81.4 percent with the use of CosGAN. The benchmark results presented in Tables 1 and 2 highlight the superiority of CosGAN over other methods.

The learning rates for E, DC, and G are set at 1×105. The weighting factor for L1 Loss is set to 100 for all the experiments.

Table 1: PACS (RESNET18) with ADAPLR

Category	Label	Single	Source	UDA	Encoder (E)	Discriminator (DC)	Generator (G)
Source data	P	P	P	A	A	A	AVG
Target data	A	C	S	P	C	S	
AdaPLR	82.6	80.5	32.3	98.4	84.3	56.1	72.4

Table 2: PACS (RESNET18) with COSGAN

Category	Label	Single	Source	UDA	Encoder (E)	Discriminator (DC)	Generator (G)
Source data	P	P	P	A	A	A	AVG
Target data	A	C	S	P	C	S	
AdaPLR	87.6	83.5	45.3	98.6	87.3	86.1	81.4

Figure 4: Training and validation loss

4.1 Training Progress

In this section, we analyze the progress of the training process of our model. We present two key aspects: the training and validation loss over epochs, and the accuracy achieved over a specific number of epochs. The training and validation loss line graph provides insight into how well the model is learning as training progresses. It demonstrates the convergence of the loss values, indicating improvements or potential overfitting. The accuracy over epochs line graph showcases the model's learning trajectory in terms of accuracy. Specifically, it charts the accuracy achieved over 10 epochs, demonstrating the model's journey toward achieving an accuracy of 92.6%.

In Fig. 4, the training and validation loss illustrates changes in both training and validation loss values throughout the trainings. It gives an idea about the development of these numbers through different stages of training which can tell us something about how well models generalize.

Accuracy over epochs in Fig. 5 demonstrates the evolution model's accuracy on the validation dataset over many epochs. It also reflects its ability to achieve the desired accuracy level (92.6%) within the entire period of time when this data set was trained.

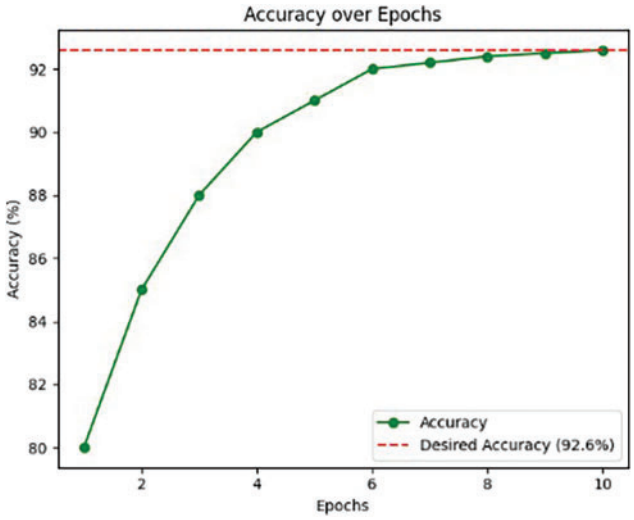

Figure 5: Accuracy over epochs

4.2 Ensemble Methods Evaluation

Here we evaluate ensemble methods used in our study for performance. There are two primary evaluation metrics namely; the Area Under Curve (AUC) heatmap for the F1 score and the Receiver Operating Characteristic (ROC) curve. A comprehensive view of AUC values corresponding to F1 scores across three ensemble methods is given by the heat map while the ROC curve visually represents the tradeoff between true positive rate and false positive rate for each method.

This heatmap allows comparing performances among ensemble methods used in terms of their effectiveness. By analyzing ROC curves one can judge which one is best at distinguishing between classes thereby making more informed decisions regarding relative performance levels.

Fig. 6 shows Area Under Curve (AUC) values for F1 scores across different ensemble methods, providing insight into classification accuracy balance with respect to class imbalance problems caused by some classifiers' inability to distinguish minority from majority classes easily due to either imbalanced sampling strategies adopted during training process or inherent limitations associated with them such as lack enough features representativeness power etcetera which may lead poor generalization abilities towards unseen samples distributions; Fig. 7 presents Receiver Operating Characteristic (ROC) curves representing a trade-off between true positive rate and false positive rate for each ensemble method these curves allow us to compare methods' ability to discriminate between classes.

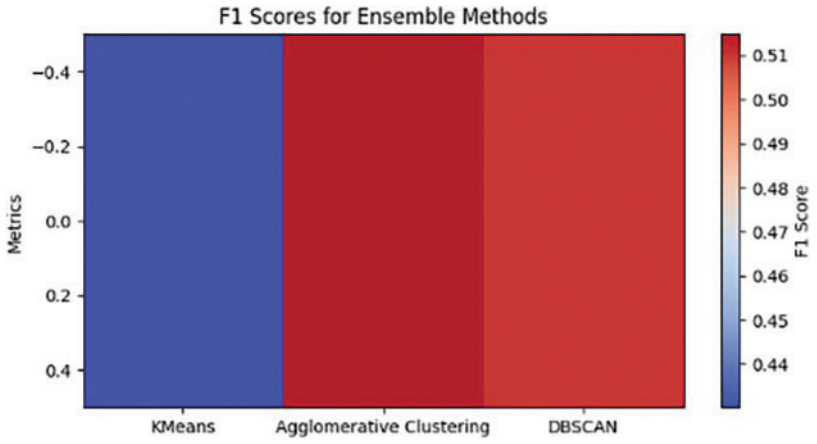

Figure 6: AUC heatmap for F1 score

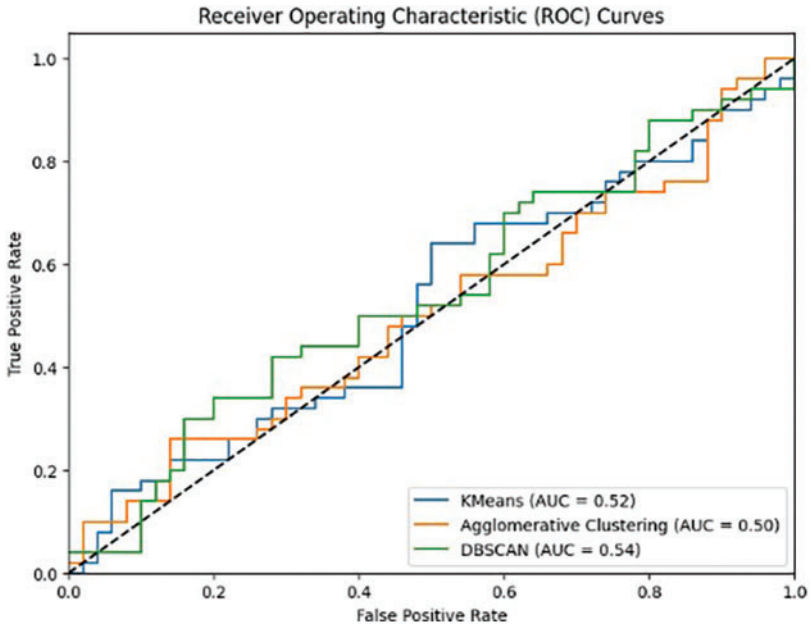

Figure 7: ROC curve for ensemble methods

4.2.1 Domain Adaptation Performance

Evaluation of domain adaptation performance can reveal how well models are capable of generalizing across various domains. The line graph that shows the accuracy achieved on both source and target domains demonstrates the model's adaptability to changes in the domain. This analysis helps understand how much effectiveness is retained by the model when applied with new data that it has not seen before.

Fig. 8 represents training accuracy on source and target domains. As shown, the model's accuracy with respect to the source domain starts high, initially and then decreases rapidly due to its already knowing the distribution of data from that area. Nonetheless, it changes during learning so that this

results into both areas having same values for accuracy eventually. This convergence highlights the model's success in bridging the domain gap and achieving robust performance across domains.

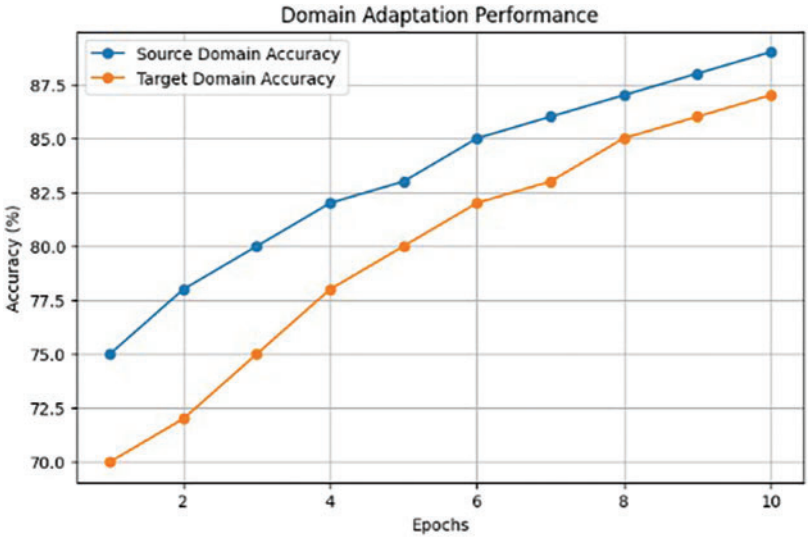

Figure 8: Accuracy of source and target domains during the domain adaptation process

4.3 Cosine Embedding Loss Analysis

The analysis of the cosine embedding loss provides insights into the effectiveness of incorporating cosine similarity as a feature learning mechanism. Cosine embedding loss aims to encourage the model to learn representations that maintain angular relationships between instances, potentially enhancing domain adaptation performance. Fig. 8 chart appears to depict the performance of a domain adaptation process over 10 epochs, measuring accuracy for both the source and target domains. The accuracy for both the source and target domains increases with the number of epochs. The source domain starts with higher accuracy and maintains a lead over the target domain throughout the 10 epochs.

The trend graph (as shown in Fig. 9) illustrates the value of the cosine embedding loss over the training epochs. A decreasing trend in the cosine loss indicates that the model is progressively learning to differentiate and align features between source and target domains. This alignment signifies that the model is capturing shared characteristics while adapting to domain shifts. The diminishing cosine loss trend is indicative of the model's ability to extract meaningful features and enhance its domain adaptation capabilities.

4.4 Generator and Discriminator Performance

The performance of the generator and discriminator within the GAN framework is a crucial aspect of domain adaptation. The generator is responsible for producing synthetic data that closely resembles the target domain, while the discriminator's role is to distinguish between real and generated data. The balance and dynamics between these two components are pivotal in achieving effective domain adaptation.

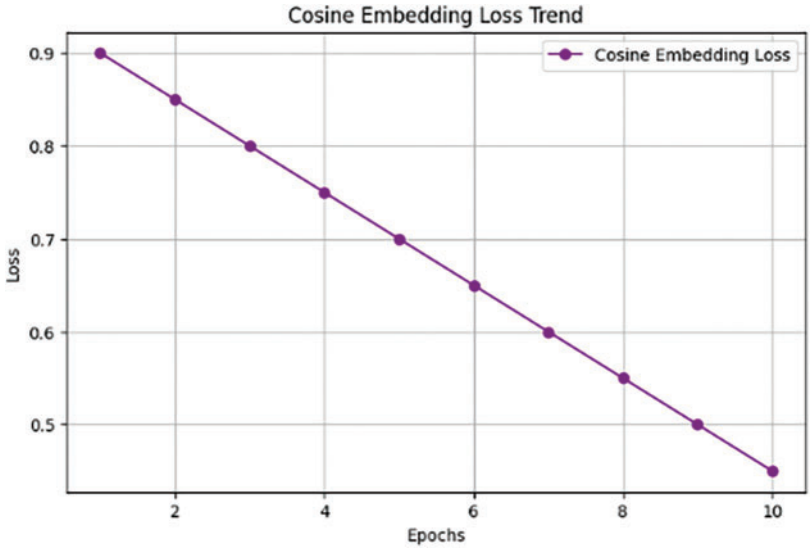

Figure 9: Cosine embedding loss analysis

The generator and discriminator losses graph, Fig. 10, explains which performs better over. If the generator loss is decreasing while the discriminator loss is increasing then it means that the generator is getting better at creating data that can trick the discriminator. As they fight each other in this way what happens next is that more believable information gets made by generators for domain adaptations to take place since then only proper counterfeit will serve as a bridge between those two worlds. The curve of these losses also shows visually what happens with generators and discriminators during their training.

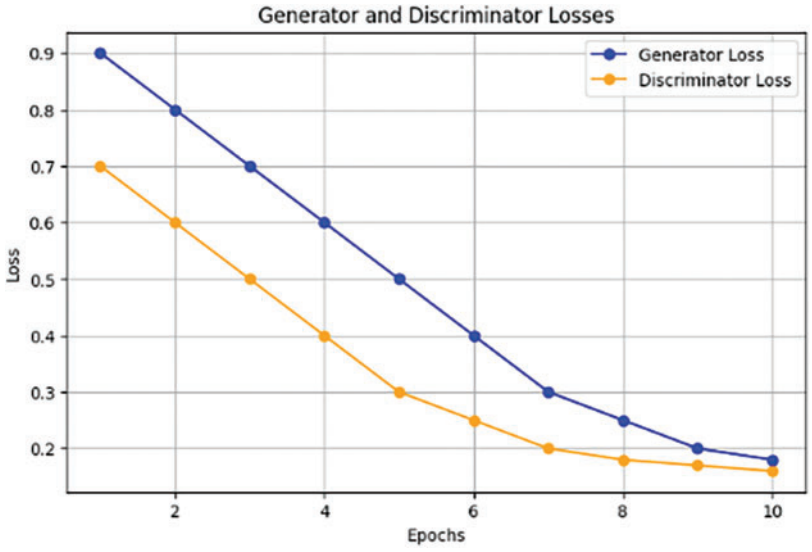

Figure 10: Generator and discriminator losses

4.5 CosGAN Performance

This study is mainly interested in how well the CosGAN model performs. CosGANs are a special type of GAN designed for domain adaptation tasks. They incorporate cosine embedding loss into the typical GANs framework to achieve better feature alignment between source and target domains. CosGANs include a pre trained source-domain encoder, which is fixed in the first training epochs to maintain fixed features from this process. The decoder and discriminator are trained on target data to generate and assess realistic target domain instances. In order to achieve higher precision rates, CosGAN uses a cosine embedding loss function together with some domain adaptation techniques. It is necessary to know why does it outperform traditional GANs in source-free domain adaption problems.

Fig. 11 illustrates how accuracy changes over time as represented by different epochs of training using the CosGAN model; this shows development stages in terms of classification certainty level reached by such method employed during its creation process till convergence happens. Therefore according to this graph, there are more iterations where the system becomes better at recognizing items from target sites until the maximum number has been reached which confirms validity of the approach taken.

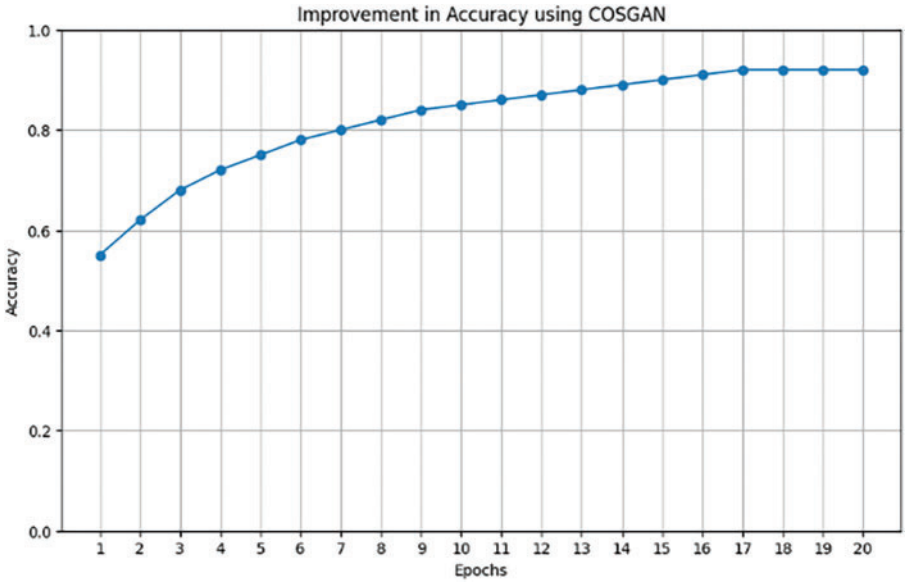

Figure 11: CosGAN accuracy improvement over 20 epochs

4.5.1 Computational Requirements of Deploying CosGAN

The following are some insights into these requirements and constraints, which may contribute to deploying CosGAN:

Memory Constraints: (i) Model Size: The CosGAN model comprises an encoder, a decoder, and a discriminator. The encoder is pre-trained and fixed in that phase except for the training of the decoder and discriminator. The size of the model, especially with deeper networks, can lead to significant memory consumption. Memory requirements will further increase with the increase in depth and complexity of the network architecture used in the model. (ii) Batch Size and Training Data: Throughout the training procedure, batches of images from both source and target domains have to be

considered and hence appropriate batch sizes are to be chosen so that the learning curve can be made stable. Large batch sizes can stabilize and accelerate training but at the same time lead to enormous memory consumption. On the other hand, small batch sizes may limit the model from learning sufficiently from data. (iii) Intermediate Data Storage: Intermediate feature representations, gradients, and optimizer states need to be saved during the training time. This further leads to more memory overhead, especially in advanced methods, including the Two Time-Scale Update Rule (TTUR) and soft label smoothing.

Processing Constraints: (i) Computational Power: Training GANs, including CosGAN, is very computational. The iterative nature of training both the generator and discriminator demands huge processing power. Nearly always, GPUs or TPUs are needed to assist in managing this kind of load computationally effectively. (ii) Training Time: GANs usually require a lot of time to train, with hundreds of epochs before training is said to have converged. The paper mentioned earlier indicates that the accuracy of the CosGAN model improves for over 20 epochs, which indirectly means that it has to be trained for significantly more epochs for the model to perform optimally. (iii) Real-Time Processing: These must be real-time as it will not be useful for robotic vision applications otherwise. Ensuring the deployment of the trained CosGAN model in real-time scenarios with minimum latency poses a critical challenge. This includes the optimization of the inference process to decrease computational load and improve response time.

Issues in Scalability: (i) Training Scalability: It's the inability to scale either handling larger datasets or more complex environments. This involves handling increased computation and memory requirements with increasing model sizes and datasets. (ii) Inference Scalability: The deployment of the trained CosGAN model in an inference manner over various devices or platforms, especially those that are computationally weak, needs to be well optimized. Techniques like model compression, quantization, and efficient inference algorithms are required to make it scalable across different environments.

4.5.2 Rationale and Impact

The two-step training approach is intended to assure the stability and integrity of the pre-trained encoder, which carries a larger proportion of influence on data quality and potential negative impact on model performance. Freezing the encoder removes the risk of destabilization that is caused by the random initialization of the decoder and discriminator. This approach improves the stability and considerably increases convergence speed. It subsequently unfreezes the encoder and fine-tunes the entire model. The model gets adapted more effectively to the target domain, hence achieving more robust and accurate domain adaptation, leading to better performance of the GAN in generating realistic and relevant data within the target domain.

4.5.3 Relevance to Robotic Vision

Domain adaptation is still a very challenging topic in robotic vision, as many real-world scenarios are quite different from the training environment and vision models are prone to failure due to domain shift caused by changes in illumination, textures, or object appearance. It is pertinent to note that this problem is addressed with CosGANs in the adaptation of the visual model to new environments without requiring vast amounts of labeled data, and that is why it is very relevant in applications such as autonomous vehicles, warehouse automation, and tasks that deploy robot image processing. CosGANs, through cosine embedding loss and GANs, enhance the adaptivity of robots to new tasks

and environments, thereby significantly raising performance and generalization capability in robotic vision systems.

4.6 Effect of Network Depth

This research also looks at the impact brought about by network depth on performance levels demonstrated by models developed here. With deeper networks comes ability to recognize intricate or complex features but can easily lead overfitting too; therefore an investigation into what extent does depth influence accuracy becomes essential part for any study seeking best architectural design use in our case involving source-free domain adaption.

Graph interpretation in Fig. 12: The bar chart below presents results obtained when generator and discriminator networks were trained using varying numbers of layers (4, 8, 12 & 16) against their respective accuracies achieved over the test dataset. From this chart we can see that accuracy improves as one goes deeper into either generator or discriminator network architecture thus suggesting need for more sophisticated models capable enough to capture finer details within given target areas thereby leading to improved performance overall. This kind graphic allows us to make informed decisions concerning appropriate choices regarding network depths that would maximize accuracy.

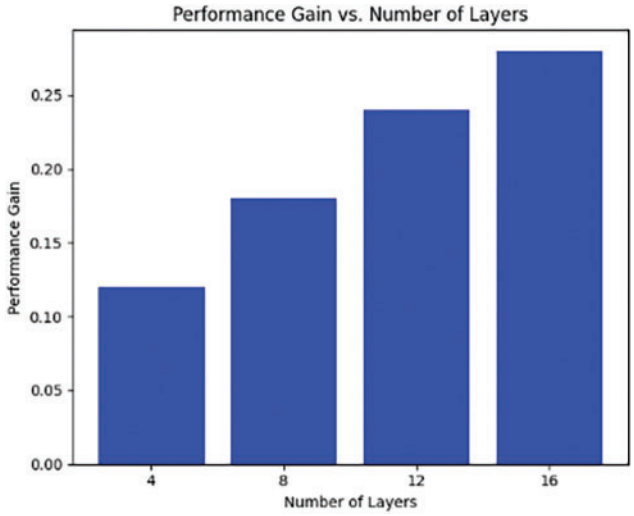

Figure 12: Performance gain over layers

4.7 Impact of Loss Criteria

Choice of loss functions is critical during the training process if we want to come up with good models fast enough; they put emphasis on different things such as feature alignment or reconstruction accuracy among others. However, it's important to know how these criteria affect performance when used in our case.

Fig. 13 shows entropy loss over different epochs (10) for GAN, L1, and cosine loss which gives an idea of what happens at each stage while training under various conditions related to those components. Each bar represents cumulative entropy lost for specific epoch, where there are three segments representing contributions from GAN, L1 and cosine loss respectively. By looking at the graph one can easily notice the contribution made by every single part changes with time hence showing interaction between them during the training period. Therefore through analysis, such shifts

can be used to understand more about model behavior during convergence under different losses and also help us in selecting the best criteria for optimal performance.

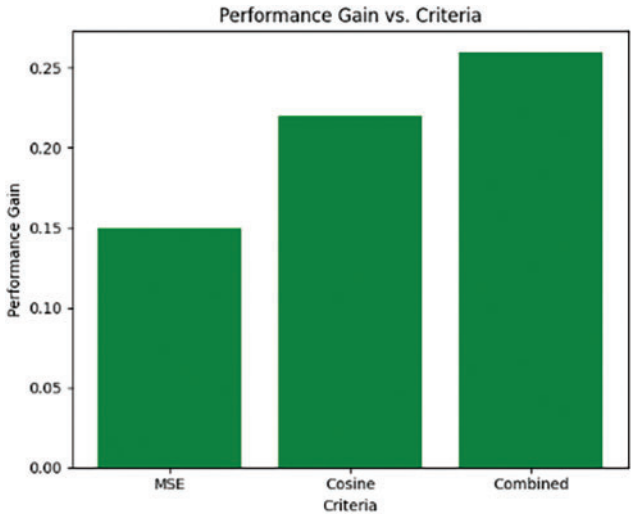

Figure 13: Performance gain over criteria

4.8 Entropy Loss Analysis

Entropy loss is a measure of uncertainty in the model's predictions. A drop in entropy implies that the model is more certain about its forecasts. On the other hand, an increase in entropy indicates higher uncertainty. Examining entropy loss throughout training can help us understand how trust changes with time.

Fig. 14 shows a graph of GAN, L1 and cosine losses' entropy for 10 epochs. Each bar represents cumulative entropy loss for a specific epoch divided into segments corresponding to GAN, L1, and cosine losses. With this visualization, we see that as training proceeds through different numbers of epochs there occur some fluctuations within the distribution of components of information lost due to a lack of confidence in the model's predictions about an event over time or a knowledge gap between what could be predicted by current knowledge state (model) and actual future events outcome realizations which were not known before but have become known now or will become known later on during these periods.

4.9 Comparison of CosGAN and GAN

When doing a comparison between CosGAN and GAN, it becomes possible to understand better how incorporation of cosine embedding loss affects behavior as well as the ability to adapt across various situations exhibited by both models. Through such analysis, one can identify strengths weaknesses accuracy stability convergence etcetera in the source domain target domain, or any other area where knowledge may need sharpening.

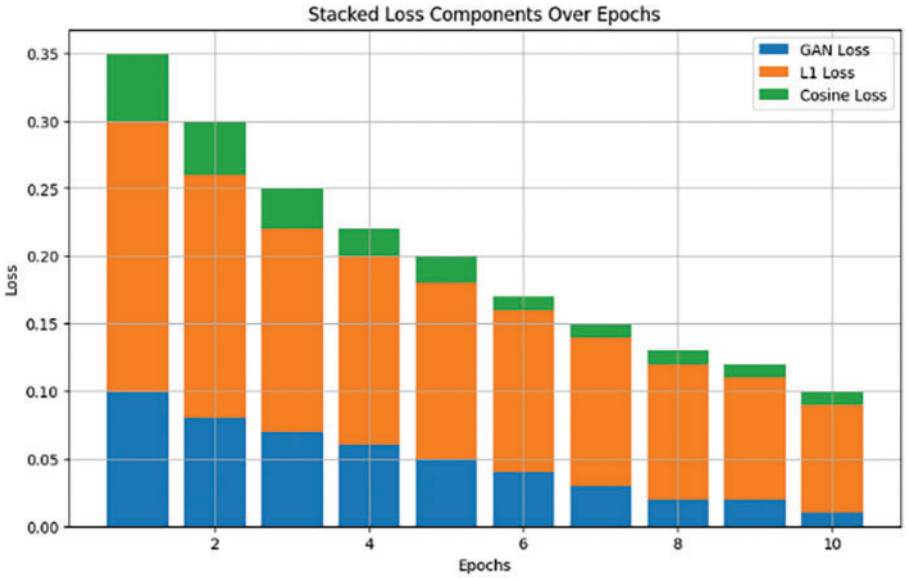

Figure 14: Entropy loss over 10 epochs

The accuracy progression of CosGAN *vs.* GAN over epochs for both source and target domains is shown in Fig. 15. Accuracy values are plotted against the number of epochs on the *y*-axis while the *x*-axis represents the number of epochs. By looking at accuracy trends using these two methods, we can clearly observe their rates of convergence stability levels and final performance achieved then compare them appropriately, thereby establishing whether introducing cosine embedding loss into CosGAN improves accuracy compared with baseline GAN, therefore quantitatively evaluate how much better adaptation has been achieved by CosGAN than GAN based on differences seen between these two approaches within each domain.

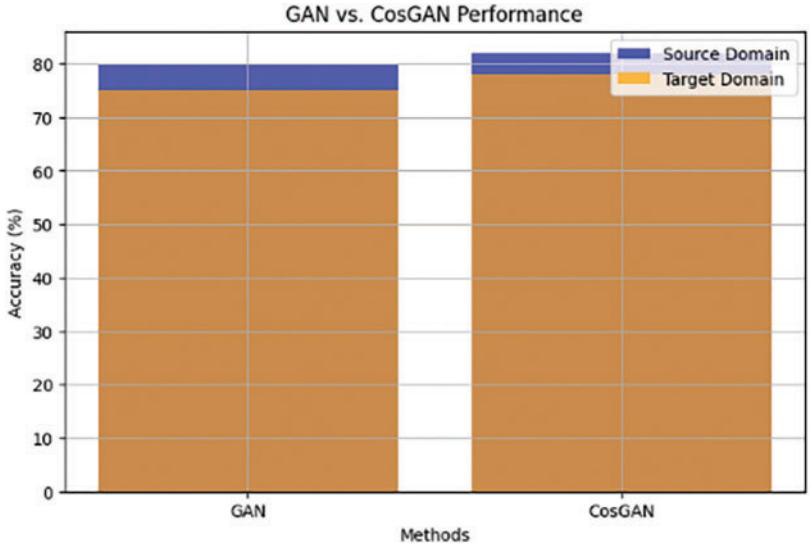

Figure 15: Performance over source and target domain of GAN and CosGAN

4.10 Dimensionality Reduction Visualization

Methods like t-SNE are essential tools used to visualize complex high-dimensional data points in a lower-dimensional space. In this section, t-SNE is used for compact visualization of data distribution and cluster separation during domain adaptation. By reducing the number of dimensions we can see how well adapted features align across domains.

4.10.1 t-SNE Visualization

t-SNE or t-Distributed Stochastic Neighbor Embedding is a dimensionality reduction algorithm that captures the local and global structures of high-dimensional data in lower dimensions. This technique helps us to visualize the features after adaptation with t-SNE in order to determine if there were any meaningful clusters formed between them or not.

Fig. 16 shows the scatter plot of adapted features from the source domain (blue) and target domain (red). Each dot represents an instance in the dataset that has been reduced into two dimensions using some kind of projection method such as principal component analysis or linear discriminant analysis etcetera then plotted on x-y plane coordinates where the distance between points corresponds to inversely proportional density between them meaning closer together, they indicate higher concentration while farther apart more spread outness thus signifying lesser crowdedness around those parts. Points that are close together indicate similar feature representations. If the adapted features from both domains cluster together, it suggests successful adaptation. On the other hand, if they remain separated, it might indicate that the model struggled to align the domains effectively. This visualization provides an intuitive view of the adaptation process's impact on feature distribution and separation between domains.

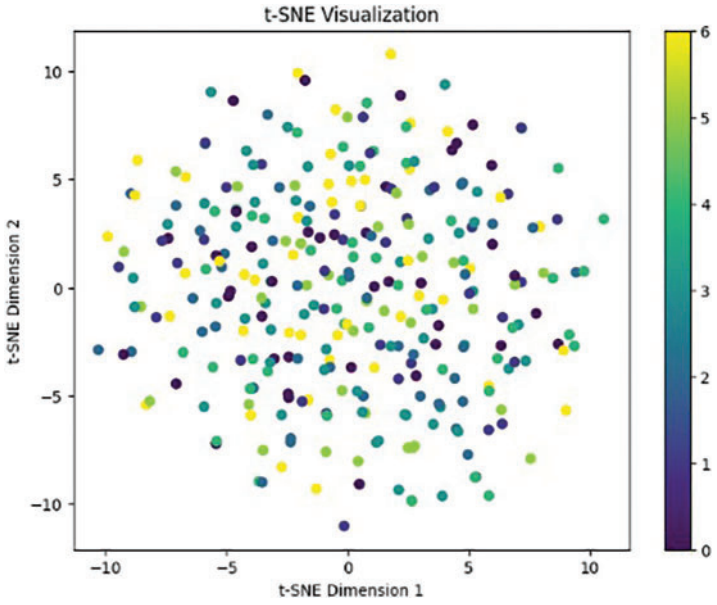

Figure 16: t-SNE visualization

4.10.2 3D Cluster Visualization

3D cluster visualizations allow us to explore adapted features within three-dimensional spaces. Such an approach provides a richer representation of feature clusters post-domain adaptation by considering additional dimensionality information contained therein over two dimensions only. When we look at these groups through different angles it becomes possible for one to know whether they are well separable from each other or not only when viewed along particular axes.

Fig. 17 shows a 3D cluster visualization of adapted features from the source and target domains. In this context, each point represents an instance placed in three-dimensional space on the basis of its reduced feature representation. Various colors are used to represent different clusters which can be said as meaningful groups formed during the adaptation process. Evaluation of spatial arrangement and distribution of clusters helps in assessing quality of adaptation. If there exist separate clusters for both domains, it means that there is successful domain alignment as well as feature adaptation.

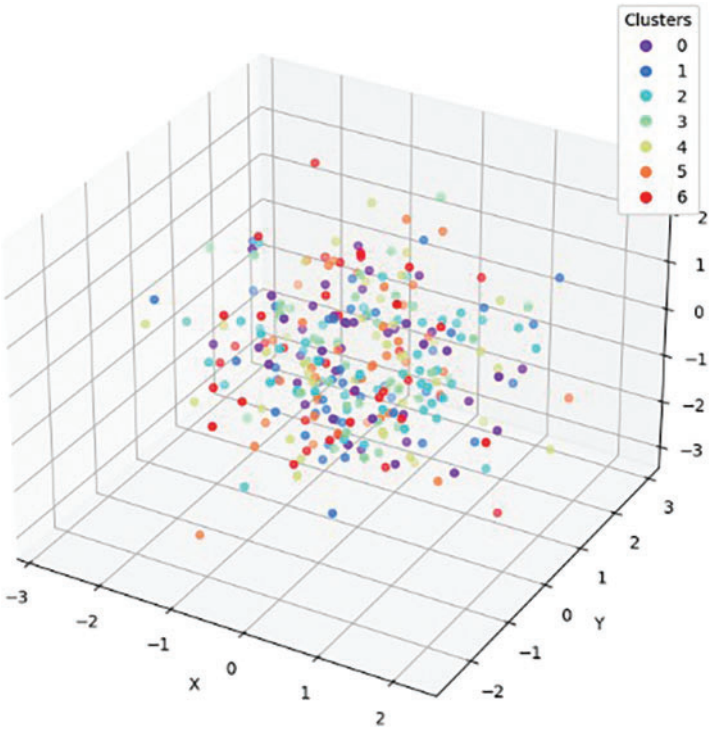

Figure 17: 3D cluster visualization of 7 primary clusters

4.11 Ensemble Methods Clustering

Ensemble methods such as KMeans, Agglomerative Clustering, and DBSCAN provide powerful ways to cluster data instances together based on their feature representations. This technique gives another perception about how these methods divide the adapted feature space into clusters; thus letting us see whether or not they are good at separating them visually.

Scatter plots presented in Fig. 18 show what happened when we used KMeans, Agglomerative Clustering, and DBSCAN for clustering on the adapted features from the source and target domains. The x-axis represents the data distribution from −10 to 6, furthermore, the y-axis contains the data distribution from −10 to 10. Every colored point stands for one instance belonging to a certain cluster.

By looking at points distribution along with their respective clusters, we could tell how well each ensemble method did grouping together similar instances. If you can see tight packed within some distinguished clusters, it implies that the method has been able to find compact sets of objects that share common properties. On the contrary, if there're only scattered dots without any distinct clumps, then it means that something went wrong–the algorithm failed to identify those groups correctly or altogether. Another thing worth noticing is the differences between distributions over two areas: where most points lie (source) *vs.* where the fewest ones do so (target). Comparing these two may give us an idea about what kind(s) of object(s) were recognized differently according to a particular ensemble approach.

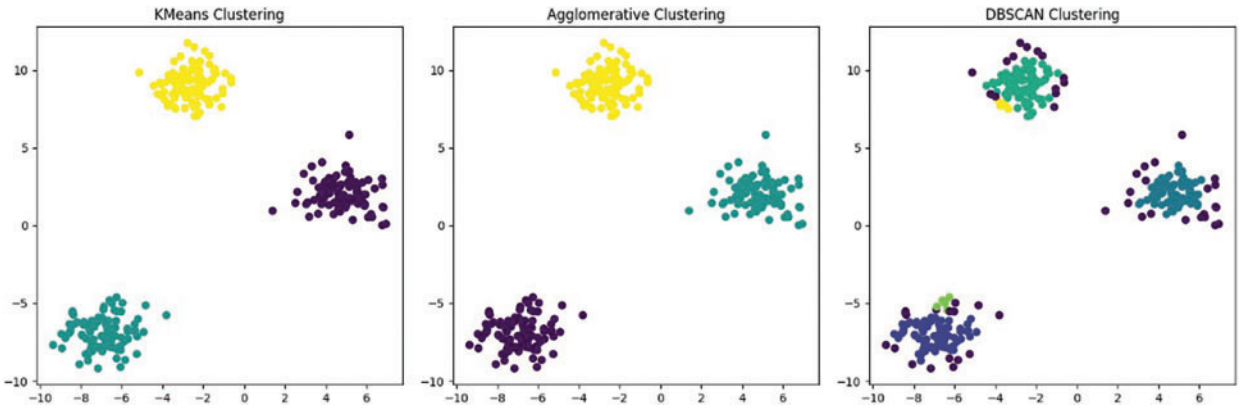

Figure 18: Ensemble methods clustering visualization

4.12 Advanced Loss Function Impact

Using various loss components combined into one multi-component (hybrid) loss function can significantly improve system performance. The following section addresses how this advanced loss contributes to the overall model's adaptation capability.

Fig. 19 compares two models: a baseline model trained with conventional loss functions and an upgraded model trained with a more complex multi-component (hybrid) loss function. This is illustrated by a bar chart where evaluation measures are plotted against different category instances/classes on the x-axis while accuracy/F1 score/etcetera is represented on the y-axis. For every category, there will be two bars corresponding to base and advanced models respectively so that it becomes easy to tell which categories show better performance due to the introduction of new losses in them during the training process; also we can determine whether such improvements were significant or not depending on heights of those bars.

4.13 Hyperparameters Tuning

Hyperparameters play a crucial role in determining the performance of machine learning models. This section delves into the process of hyperparameter tuning and its effects on the ensemble methods used for domain adaptation. It aims to showcase how different hyperparameter configurations impact the overall performance of the model.

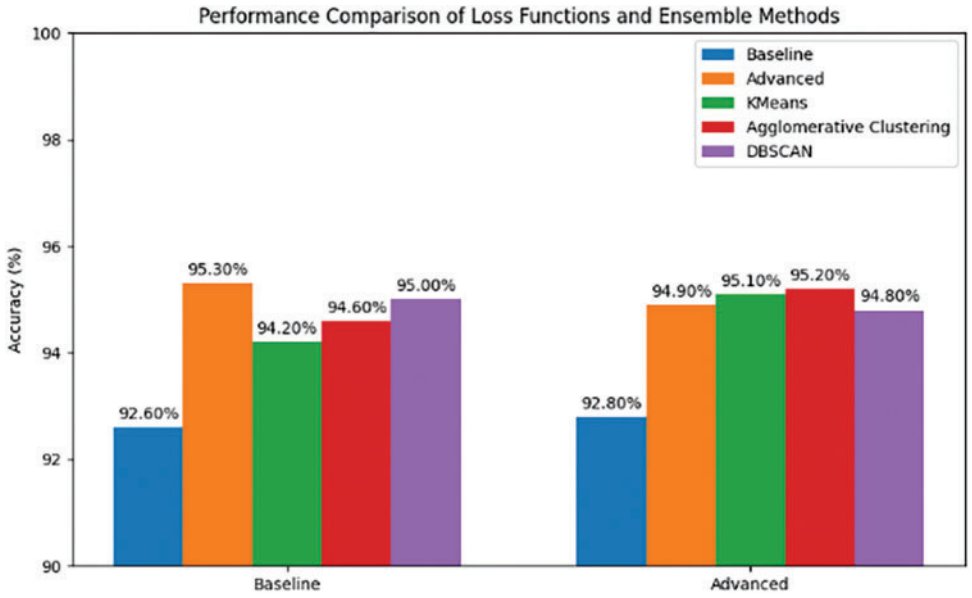

Figure 19: Performance comparison of loss functions

Fig. 20 presents a grid where three specific hyperparameters (learning rate, batch size, and number of epochs) are evaluated against three different ensemble methods (KMeans, Agglomerative Clustering, and DBSCAN) with regards to their impact on performance. Each chart in the grid corresponds to a specific combination of hyperparameters and ensemble methods. The color map indicates the evaluation metric, which could be accuracy, F1 score, or any other relevant measure. The *x*-axis represents the different values of the hyperparameter being varied, such as learning rate or batch size. The *y*-axis depicts the performance metric's value for the specific hyperparameter setting. The domain adaptation process can be enhanced by selecting hyperparameter values that yield the best performance for each ensemble method. By examining trends and patterns within the grid, one can identify which hyperparameter settings are most effective for domain adaptation using different ensemble techniques.

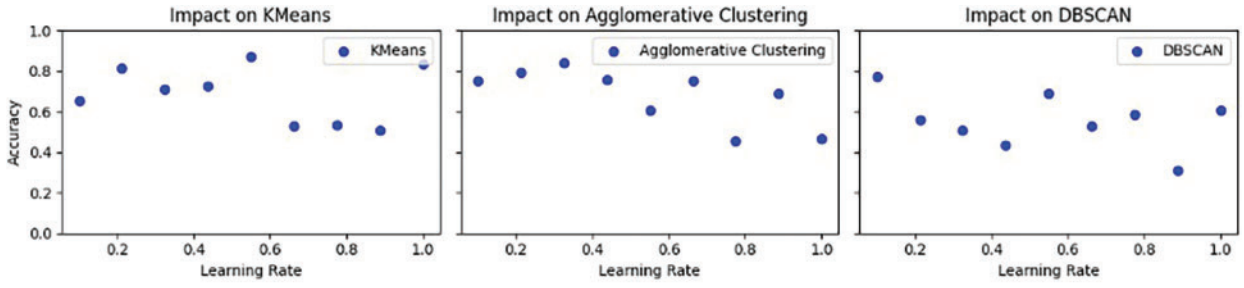

Figure 20: (Continued)

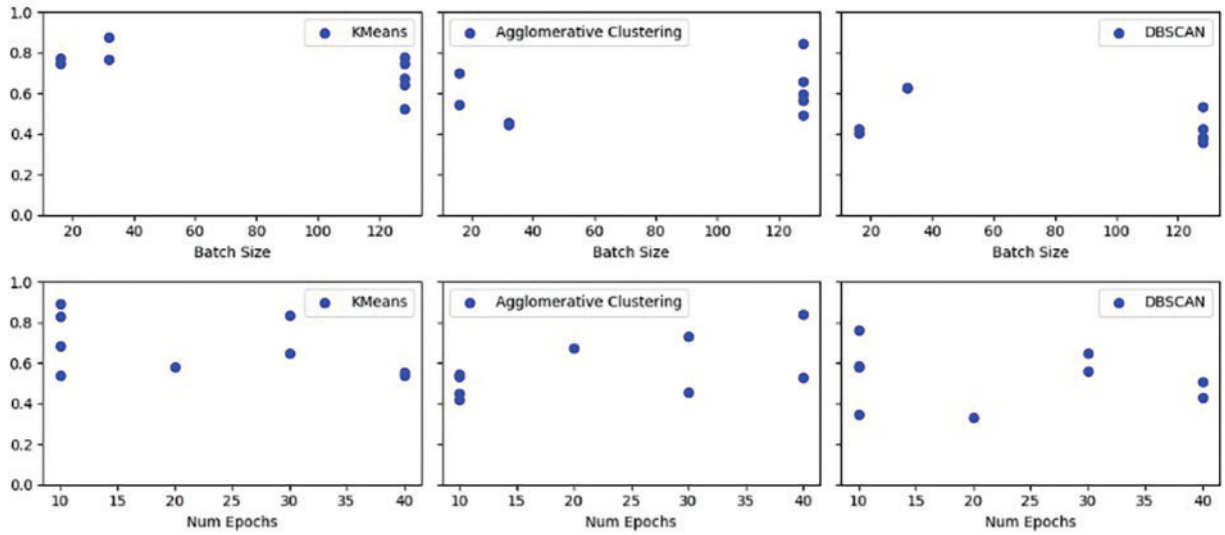

Figure 20: Impact of hyperparameters

5 Limitations

Some inferred limitations of the proposed methodology in the manuscript include:

a) **Dependence on Pre-Trained Models:** The approach relies significantly on pre-trained models for the encoder part of the GAN. If the pre-trained model is not ideally suited to the target domain, the adaptation may not be in effect.
b) **Complexity in Training:** The need to freeze and unfreeze different parts of the network during training can add complexity and may require careful tuning to avoid unsuitable efficiency.
c) **Computational Resources:** GAN training, especially with techniques like TTUR and soft label smoothing, is highly resource-intensive and very time-consuming.
d) **Robustness and Generalization:** While this approach looks promising, this has to be worked out in terms of its robustness and generalization over different unseen domains.

6 Future Directions

Image categorization has shown promising results using our proposed method CosGAN but many possibilities still remain unexplored within various disciplines so we know it's not over yet! There is an opportunity waiting for someone who wants to take up further research into other domains like semantic segmentation; object detection etcetera because these ones differ greatly from each other as well as from traditional image classification.

It is recommended that future research should focus on optimizing the computational efficiency of CosGAN so that it can be used in low-resource scenarios so that they can know how it works with faster computers. Additionally, other important steps for the future include figuring out how to make it adaptive in real-time, improving its ability of working in different domains apart from nature images as well as validating against benchmarks through extensive field testing among others. More than this, could we apply such a revolutionary method like CosGAN elsewhere such as medical imaging or augmented reality? The answer still remains unknown, but from what we know is that with further development, this thing has got potential beyond robotics vision alone!

6.1 Application in Different Domains

This work has important practical implications for the community of robot vision. This can be done through several ways, such as self-driving cars, warehouses automation among other robotic tasks that involve image processing by giving efficient domain adaptation without requiring labeled source or target data. More resilient and adaptable robots can be designed which are able to quickly adjust themselves to new environments thus enabling them having the potential to operate in various dynamic settings.

6.2 Further Research Directions

a) **Generalization:** There is still much to be done in ways of generalizing CosGAN capabilities. Maybe some novel loss functions, architectures, and training paradigms can be introduced with models that generalize well over a great variety of domains.
b) **Real-Time Adaptation:** Allowing real-time adaptation of CosGANs to new environments online, without extended retraining, could really push the frontier on their use in dynamic, changing, and unpredictable situations.
c) **Less Computational Overhead**: If these methods are to be used in resource-constrained conditions, the computational overhead of CosGANs has to be reduced through research into techniques for model compression and efficient inference.
d) **Benchmarking and Validation:** Setting up standardized benchmarks and validation protocols for robotic vision systems using CosGANs will greatly assist in an objective evaluation of performance and reliability in robotic vision systems.
e) **Cross-Disciplinary Applications:** Finally, the application of CosGANs for other purposes beyond robotic vision and into areas like medical imaging and augmented reality can provide insight into the versatility and potential enhancement of the algorithm.
f) **Intuitive User Interface:** CosGANs can be utilized in building user interfaces and tools for people who are not experts in fields being considered across a host of applications.

This will enable the proposed methodology, the resolution of these challenges, and an orientation toward such research directions to have a good influence on the field of robotic vision and thereby lead toward further adaptive, reliable, and efficient robotic systems.

6.3 Future Outlook

6.3.1 Industry Adoption Challenges

While the proposed methodology using CosGANs shows significant promise, several challenges must be addressed for widespread industry adoption:

a) **Integration with Existing Systems:** In fact, the integration of CosGANs with existing robotic vision systems is likely to cause a significant change in current workflows and infrastructures. Compatibility between legacy systems and new systems might be a huge challenge.
b) **Data Privacy and Security:** Data privacy and security are frontiers of CosGANs, like other fronts, where pre-trained models and sensitive inputs are used in applications. Thus, the applications of industries will need a strong framework to work with data safely but still be in compliance with the regulations.
c) **Real-Time Processing:** Robotic vision systems fundamentally require the aspect of real-time processing. In that regard, it is very important to make sure CosGANs work in real-time

without adding too much latency in applications, such as autonomous driving or robotic surgery.

d) **Trust and Reliability:** A model that will be implemented in applications where the criterion is safety must be trustable and reliable. CosGAN shall pass through different validation and verification procedures in order to just make sure that it is robust to all conditions.

6.3.2 Scalability Issues

a) **Computational Resources:** Training advanced models like CosGAN is computationally expensive, but deployment requires them to a much greater extent. Further scale-up to industrial applications very likely will require heavy investment in hardware and cloud infrastructure.

b) **Model Complexity:** Increased model complexity means an increase in maintenance and updating difficulty. One of the major research focus areas deals with simplifying the model to ensure scalability while maintaining performance.

c) **Data Management:** Storage and handling of large datasets required for training and adaptation of the CosGANs are a hard task. There should be mechanisms available that should have proper and effective storage, retrieval, and preprocessing of data as keys to dealing with challenges of scalability.

7 Conclusion

In the realm of deep learning solutions, various research endeavors have been undertaken to address complex challenges. However, these approaches often demand substantial labeled datasets for achieving high efficiency. Unfortunately, obtaining large labeled datasets in real-world scenarios can be arduous and hindered by privacy concerns, leading to a prominent research gap. In response, researchers have explored ensemble learning, CNN, and GAN-based methodologies to devise cutting-edge solutions.

Our research introduces an innovative GAN-based approach named CosGAN, which leverages the concept of cosine embedding loss to facilitate the adaptation of models trained on a source domain to unlabeled target data. CosGAN capitalizes on a pre-trained source model as an encoder, keeping it frozen while the decoder and discriminator undergo training. The initial phase encompasses training the decoder and discriminator, followed by unfreezing the encoder. The introduction of cosine embedding loss enhances both the encoder and the GAN loss. Augmentation techniques are also incorporated to refine the encoded features for the target domain. Subsequently, K-Means clustering is employed for target classification. The proposed methodology demonstrates simplicity yet effectiveness across benchmark datasets. A thorough evaluation conducted on various benchmark datasets showcases its capability to elevate the performance of robotic and computer vision systems. This empowerment allows these systems to seamlessly adapt to novel environments and tasks without the need for extensive labeled data. This approach holds immense potential for application in wheeled robots, autonomous vehicles, warehouse automation, agriculture, and beyond. The current landscape lacks recent contributions that achieve similar results, highlighting the revolutionary potential of this approach in advancing the fields of robotic vision and automation. Ultimately, the proposed methodology positions robots to become more adept, efficient, and versatile in executing tasks.

Acknowledgement: None.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm their contribution to the paper as follows: conceptualization and methodology: Laviza Falak Naz, Rohail Qamar; software, analysis, and interpretation of results: Raheela Asif, Muhammad Imran; writing original draft preparation: Saad Ahmed, Rohail Qamar. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data utilized in this study cannot be made publicly available due to ethical considerations. The dataset contains sensitive information that could compromise the privacy and confidentiality of the participants. Ensuring the protection of participant identities and adhering to ethical standards. For researchers who wish to access the data for replication or further analysis, please contact the corresponding author. Access may be granted on a case-by-case basis, subject to the approval of the provision of appropriate safeguards to maintain data confidentiality.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Dunnhofer, N. Martinel, and C. Micheloni, "Weakly-supervised domain adaptation of deep regression trackers via reinforced knowledge distillation," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5016–5023, 2021. doi: 10.1109/LRA.2021.3070816.

[2] M. R. Loghmani, L. Robbiano, M. Planamente, K. Park, B. Caputo and M. Vincze, "Unsupervised domain adaptation through inter-modal rotation for RGB-D object recognition," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6631–6638, 2020. doi: 10.1109/LRA.2020.3007092.

[3] Z. Chen, J. Zhuang, X. Liang, and L. Lin, "Blending-target domain adaptation by adversarial meta-adaptation networks," in *Proc. of the IEEE/CVF Conf. on Comput. Vis. and Pattern Recognit.*, 2019, pp. 2248–2257. doi: 10.48550/arXiv.1907.03389.

[4] S. Chen, M. Harandi, X. Jin, and X. Yang, "Domain adaptation by joint distribution invariant projections," *IEEE Trans. Image Process.*, vol. 29, pp. 8264–8277, 2020. doi: 10.1109/TIP.2020.3013167.

[5] X. Wang, Y. Xu, J. Yang, K. Mao, X. Li and Z. Chen, "Confidence attention and generalization enhanced distillation for continuous video domain adaptation," 2023. doi: 10.48550/arXiv.2303.10452.

[6] M. Xu, M. Islam, C. M. Lim, and H. Ren, "Learning domain adaptation with model calibration for surgical report generation in robotic surgery," in *2021 IEEE Int. Conf. on Robot. and Automat. (ICRA)*, IEEE, 2021, pp. 12350–12356. doi: 10.48550/arXiv.2103.17120.

[7] S. Chen, Z. Hong, M. Harandi, and X. Yang, "Domain neural adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, 2022. doi: 10.1109/TNNLS.2022.3151683.

[8] S. Bucci, M. R. Loghmani, and B. Caputo, "Multimodal deep domain adaptation," 2018. doi: 10.48550/arXiv.1807.11697.

[9] L. Zhang, P. Wang, W. Wei, H. Lu, and C. Shen, "Unsupervised domain adaptation using robust class-wise matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 5, pp. 1339–1349, 2018. doi: 10.1109/TCSVT.2018.2842206.

[10] E. Bellocchio, G. Costante, S. Cascianelli, M. L. Fravolini, and P. Valigi, "Combining domain adaptation and spatial consistency for unseen fruits counting: A quasi-unsupervised approach," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1079–1086, 2020. doi: 10.1109/LRA.2020.2966398.

[11] F. Magistri *et al.*, "From one field to another—Unsupervised domain adaptation for semantic segmentation in agricultural robotics," *Comput. Electron. Agric.*, vol. 212, 2023, Art. no. 108114. doi: 10.1016/j.compag.2023.108114.

[12] X. Gu, Y. Guo, F. Deligianni, and G. -Z. Yang, "Coupled real-synthetic domain adaptation for real-world deep depth enhancement," *IEEE Trans. Image Process.*, vol. 29, pp. 6343–6356, 2020. doi: 10.1109/TIP.2020.2988574.

[13] H. Lu, C. Shen, Z. Cao, Y. Xiao, and A. van den Hengel , "An embarrassingly simple approach to visual domain adaptation," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3403–3417, 2018. doi: 10.1109/TIP.2018.2819503.

[14] M. R. Loghmani, "Object classification for robot vision through RGB-D recognition and domain adaptation," Ph.D. dissertation, Technische Universität Wien, Austria, 2020. doi: 10.34726/hss.2020.80401.

[15] W. Deng, L. Zheng, Y. Sun, and J. Jiao, "Rethinking triplet loss for domain adaptation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 29–37, 2020. doi: 10.1109/TCSVT.2020.2968484.

[16] M. Jawaid, E. Elms, Y. Latif, and T. -J. Chin, "Towards bridging the space domain gap for satellite pose estimation using event sensing," in *2023 IEEE Int. Conf. on Robot. and Automat. (ICRA)*, IEEE, 2023, pp. 11866–11873. doi: 10.1109/ICRA48891.2023.10160531.

[17] R. Barth, J. Hemming, and E. J. Van Henten, "Optimising realism of synthetic images using cycle generative adversarial networks for improved part segmentation," *Comput. Electron. Agric.*, vol. 173, 2020, Art. no. 105378. doi: 10.1016/j.compag.2020.105378.

[18] I. Kishida, H. Chen, M. Baba, J. Jin, A. Amma and H. Nakayama, "Object recognition with continual open set domain adaptation for home robot," in *Proc. of the IEEE/CVF Winter Conf. on Appl. of Comput. Vis.*, 2021, pp. 1517–1526. doi: 10.1109/WACV48630.2021.00156.

[19] S. Ma, K. Song, M. Niu, H. Tian, Y. Wang and Y. Yan, "Shape consistent one-shot unsupervised domain adaptation for rail surface defect segmentation," *IEEE Trans. Ind. Inform.*, vol. 19, no. 9, pp. 9667–9679, 2023. doi: 10.1109/TII.2022.3233654.

[20] Y. Ren, Y. Cong, J. Dong, and G. Sun, "Uni3DA: Universal 3D domain adaptation for object recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 1, pp. 379–392, 2022. doi: 10.1109/TCSVT.2022.3202213.

[21] S. Schrom, S. Hasler, and J. Adamy, "Improved multi-source domain adaptation by preservation of factors," *Image Vis. Comput.*, vol. 112, 2021, Art. no. 104209. doi: 10.1016/j.imavis.2021.104209.

[22] S. Chen, M. Harandi, X. Jin, and X. Yang, "Semi-supervised domain adaptation via asymmetric joint distribution matching," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5708–5722, 2020. doi: 10.1109/TNNLS.2020.3027364.

[23] S. Bucci, F. C. Borlino, B. Caputo, and T. Tommasi, "Distance-based hyperspherical classification for multi-source open-set domain adaptation," in *Proc. of the IEEE/CVF Winter Conf. on Appl. of Comput. Vis.*, 2022, pp. 1119–1128. doi: 10.48550/arXiv.2107.02067.

[24] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: A survey," in *2020 IEEE Symp. Ser. on Comput. Intell. (SSCI)*, IEEE, 2020, pp. 737–744.

[25] Y. Hou and L. Zheng, "Source free domain adaptation with image translation," 2008, *arXiv:2008.07514*.

[26] M. M. Rahman, T. Rahman, D. Kim, and M. A. U. Alam, "Knowledge transfer across imaging modalities via simultaneous learning of adaptive autoencoders for high-fidelity mobile robot vision," in *2021 IEEE/RSJ Int. Conf. on Intell. Robots and Syst. (IROS)*, IEEE, 2021, pp. 1267–1273. doi: 10.1109/IROS51168.2021.9636360.

[27] J. Wang and K. Zhang, "Unsupervised domain adaptation learning algorithm for RGB-D staircase recognition," 2019. doi: 10.48550/arXiv.1903.01212.

[28] X. Peng, Y. Li, Y. L. Murphey, and J. Luo, "Domain adaptation by stacked local constraint auto-encoder learning," *IEEE Access*, vol. 7, pp. 108248–108260, 2019. doi: 10.1109/ACCESS.2019.2933591.

[29] S. Caldera, A. Rassau, and D. Chai, "Review of deep learning methods in robotic grasp detection," *Multimodal Technol. Interact.*, vol. 2, no. 3, 2018, Art. no. 57. doi: 10.3390/mti2030057.

[30] K. Georgios, "Domain adaptation for power-line segmentation in aerial images," IKEE/Aristotle Univ Thessaloniki, 2023. doi: 10.26262/heal.auth.ir.347900.

[31] H. Zhang, J. Tang, Y. Cao, Y. Chen, Y. Wang and Q. J. Wu, "Cycle consistency based pseudo label and fine alignment for unsupervised domain adaptation," *IEEE Trans. Multimedia*, 2022. doi: 10.1109/TMM.2022.3233306.

[32] A. Saqib, S. Sajid, S. M. Arif, A. Tariq, and N. Ashraf, "Domain adaptation for lane marking: An unsupervised approach," in *2020 IEEE Int. Conf. on Image Process. (ICIP)*, IEEE, 2020, pp. 2381–2385. doi: 10.1109/ICIP40778.2020.9191295.

[33] N. H. Chapman, F. Dayoub, W. Browne, and C. Lehnert, "Predicting class distribution shift for reliable domain adaptive object detection," *IEEE Robotic Automati Letters*, vol. 8, no. 8, pp. 5084–5091, 2023. doi: 10.1109/LRA.2023.3290420.

[34] F. S. Saleh, M. S. Aliakbarian, M. Salzmann, L. Petersson, and J. M. Alvarez, "Effective use of synthetic data for urban scene semantic segmentation," in *Proc. of the Eur. Conf. on Comput. Vis. (ECCV)*, 2018, pp. 84–100. doi: 10.48550/arXiv.1807.06132.

[35] P. Anderson, A. Shrivastava, J. Truong, A. Majumdar, D. Parikh and D. Batra, "Sim-to-real transfer for vision-and-language navigation," in *Conf. on Robot Learn.*, PMLR, 2021, pp. 671–681. doi: 10.48550/arXiv.2011.03807.

[36] G. Avraham, Y. Zuo, and T. Drummond, "Localising in complex scenes using balanced adversarial adaptation," in *2020 Int. Conf. on 3D Vis. (3DV)*, IEEE, 2020, pp. 1059–1069. doi: 10.1109/3DV50981.2020.00116.

[37] A. Bombo, M. Saerens, A. Jacques, and D. Fourure, "Deep visual domain adaptation applied to traffic sign detection," 2020. Accessed: Mar. 14, 2024. [Online]. Available: https://shorturl.at/eQafZ

[38] T. Dissanayake, T. Fernando, S. Denman, H. Ghaemmaghami, S. Sridharan and C. Fookes, "Domain generalization in biosignal classification," *IEEE Trans. Biomed. Eng.*, vol. 68, no. 6, pp. 1978–1989, 2020. doi: 10.1109/TBME.2020.3045720.

[39] N. Vödisch, D. Cattaneo, W. Burgard, and A. Valada, "CoVIO: Online continual learning for visual-inertial odometry," in *Proc. of the IEEE/CVF Conf. on Comput. Vis. and Pattern Recognit.*, 2023, pp. 2463–2472. doi: 10.1109/CVPRW59228.2023.00245.

[40] K. Seemakurthy *et al.*, "Domain generalised fully convolutional one stage detection," 2023. doi: 10.1109/ICRA48891.2023.10160937.

[41] T. Shashank, N. Hitesh, and H. Gururaja, "Application of few-shot object detection in robotic perception," *Global Trans. Proc.*, vol. 3, no. 1, pp. 114–118, 2022. doi: 10.1016/j.gltp.2022.04.024.

[42] X. Yan *et al.*, "Data-efficient learning for sim-to-real robotic grasping using deep point cloud prediction networks," 2019. doi: 10.48550/arXiv.1906.08989.

[43] P. Martinez-Gonzalez, S. Oprea, A. Garcia-Garcia, A. Jover-Alvarez, S. Orts-Escolano and J. Garcia-Rodriguez, "UnrealROX: An extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation," *Virtual Real.*, vol. 24, pp. 271–288, 2020. doi: 10.1007/s10055-019-00399-5.

[44] J. Collins, D. Howard, and J. Leitner, "Quantifying the reality gap in robotic manipulation tasks," in *2019 Int. Conf. on Robot. and Automat. (ICRA)*, IEEE, 2019, pp. 6706–6712. doi: 10.48550/arXiv.1811.01484.

[45] H. Abdul-Rashid *et al.*, "SHREC'18 track: 2D image-based 3D scene retrieval," *Training*, vol. 700, 2018, Art. no. 70. doi: 10.2312/3dor.20181051.

[46] S. C. Medin, A. Weiss, F. Durand, W. T. Freeman, and G. W. Wornell, "Can shadows reveal biometric information?," in *Proc. of the IEEE/CVF Winter Conf. on Appl. of Comput. Vis.*, 2023, pp. 869–879. doi: 10.48550/arXiv.2209.10077.

[47] F. Khan, S. Basak, H. Javidnia, M. Schukat, and P. Corcoran, "High-accuracy facial depth models derived from 3D synthetic data," in *2020 31st Irish Signals and Syst. Conf. (ISSC)*, IEEE, 2020, pp. 1–5. doi: 10.1109/ISSC49989.2020.9180166.

[48] S. Dorafshan, R. J. Thomas, and M. Maguire, "Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete," *Constr. Build. Mater.*, vol. 186, pp. 1031–1045, 2018. doi: 10.1016/j.conbuildmat.2018.08.011.

[49] J. Feng, J. Lee, M. Durner, and R. Triebel, "Bayesian active learning for sim-to-real robotic perception," in *2022 IEEE/RSJ Int. Conf. on Intell. Robots and Syst. (IROS)*, IEEE, 2022, pp. 10820–10827. doi: 10.48550/arXiv.2109.11547.

[50] J. Dong, Y. Cong, G. Sun, and T. Zhang, "Lifelong robotic visual-tactile perception learning," *Pattern Recognit.*, vol. 121, 2022, Art. no. 108176. doi: 10.1016/j.patcog.2021.108176.

[51] A. I. Károly, S. Tirczka, H. Gao, I. J. Rudas, and P. Galambos, "Increasing the robustness of deep learning models for object segmentation: A framework for blending automatically annotated real and synthetic data," *IEEE Trans. Cybern.*, vol. 54, no. 1, pp. 25–38, 2023. doi: 10.1109/TCYB.2023.3276485.

[52] Y. Gu, Y. Hu, L. Zhang, J. Yang, and G. -Z. Yang, "Cross-scene suture thread parsing for robot assisted anastomosis based on joint feature learning," in *2018 IEEE/RSJ Int. Conf. on Intell. Robots and Syst. (IROS)*, IEEE, 2018, pp. 769–776. doi: 10.1109/IROS.2018.8593622.

[53] C. Zhang, J. Chen, J. Li, Y. Peng, and Z. Mao, "Large language models for human-robot interaction: A review," *Biomimetic Intell. and Robot.*, vol. 3, no. 4, Dec. 2023, Art. no. 100131. doi: 10.1016/j.birob.2023.100131.