Classifier Guided Cluster Density Reduction for Dataset Selection

Cheng Chang* Keyu Long* Zijian Li* Himanshu Rai*

Layer 6 AI

{jason, keyu, ted, himanshu}@layer6.ai

Abstract

In this paper, we address the challenge of selecting an optimal dataset from a source pool with annotations to enhance performance on a target dataset derived from a different source. This is important in scenarios where it is hard to afford on-the-fly dataset annotation and is also the theme of the second Visual Data Understanding (VDU) Challenge. Our solution, the Classifier Guided Cluster Density Reduction (CCDR) framework, operates in two stages. Initially, we employ a filtering technique to identify images that align with the target dataset's distribution. Subsequently, we implement a graph-based cluster density reduction method, steered by a classifier that approximates the distance between the target distribution and source distribution. This classifier is trained to distinguish between images that resemble the target dataset and those that do not, facilitating the pruning process shown in Figure 1. Our approach maintains a balance between selecting pertinent images that match the target distribution and eliminating redundant ones that do not contribute to the enhancement of the detection model. We demonstrate the superiority of our method over various baselines in object detection tasks, particularly in optimizing the training set distribution on the region100 dataset. We have released our code here: https://github.com/himsR/DataCVChallenge-2024/tree/main

1. Introduction

In supervised machine learning, the performance of predictive models heavily relies on the quality and quantity of annotated data. Factors like data volume, label accuracy, and overall quality significantly influence the outcomes of deep learning models. Obtaining high-quality annotations, however, is often expensive and time-consuming.

To address data scarcity, transfer learning approaches have become prevalent [8, 33–35], where models are pre-trained on large, diverse datasets and then fine-tuned on smaller, specific datasets or used directly for inference. Yet, as machine learning tasks grow more complex, from object detection to semantic segmentation, the difference in data distributions between training and target domains can

Figure 1. This figure illustrates the image pruning process employed by our Classifier Guided Cluster Density Reduction (CCDR) framework. The right column displays images that are excluded from the final selection due to their substantial similarity to the corresponding images in the left column within the same row. A green box encases pairs of images that exhibit a high degree of visual resemblance, indicating that they originate from identical scenes. Conversely, a red box encompasses pairs of images that, despite their similarity, are derived from distinct scenes. This differentiation is crucial in our framework to ensure the retention of diverse and representative images in the final dataset while eliminating redundant or overly similar instances.

*Authors contributed equally and order is determined randomly.

undermine the model's ability to generalize. This issue is often aggravated by dataset biases arising from varied data collection conditions, such as camera types, capture times, and locations.

While domain adaptation and generalization techniques have shown effectiveness in addressing these issues, the strategic selection of datasets remains underexplored and holds potential for significant impact [52]. This paper explores this promising research area. We utilize a labeled source dataset compiled from various public datasets and aim to extract a fixed-size subset that mirrors the target image set's distribution. Following the competition requirement[1], our process trains an object detection model to identify bounding boxes, maintaining a consistent model configuration without hyper-parameter tuning.

We propose the Classifier Guided Cluster Density Reduction (CCDR) framework, which operates in two stages. In the first stage we take a similar approach as in [52] to perform clustering and then rank the clusters on their basis of similarity to the training images of the target dataset. The first stage selects clusters closely representing the target domain by employing CLIP Maximum Mean Discrepancy (CMMD) [26] and Vision Transformer (ViT) [15] latents trained on CLIP [42] to assess sample similarity. In the second stage of our process, we focus on pruning the clusters selected during the first stage, aiming to maximize the diversity of samples while operating within the limits of our training budget. To facilitate effective pruning, we construct a similarity graph using the samples chosen in the initial stage. This graph helps us identify connected components, which are clusters of samples with high similarity. For each connected component, we then deploy a classifier to select the samples that best align with the characteristics of the target set. This methodical approach ensures that we maintain a diverse and representative subset of samples, optimizing our training resources and enhancing the model's performance on the target domain.

Our novel approach constructs a similarity graph and uses a distribution classifier to eliminate redundant images, refining the source dataset to a smaller, fixed-size subset. Empirical results demonstrate that our CCDR method effectively aligns with the target data distribution and selects training samples that optimize object detection performance.

Our key contributions are:

- Utilization of CMMD with ViT latents to accurately evaluate sample congruence, facilitating the selection of representative clusters;
- Development of a similarity graph for selected clusters, enabling the extraction of unique samples through connected component analysis;

[1]https://sites.google.com/view/vdu-cvpr24/competition/

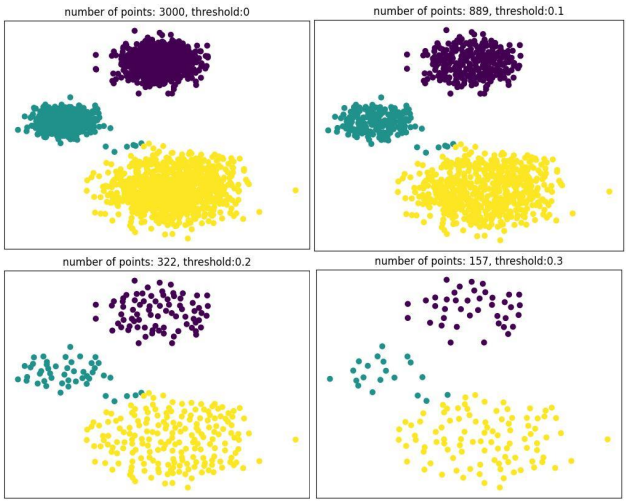

Figure 2. Number of points selected using different threshold. Top-left shows the original clusters with 0 threshold. Top-right shows threshold of 0.1, with 3 times less points. Bottom-left shows threshold of 0.2, with 10 times less points. Bottom-right shows threshold of 0.3, leaving about 20 times fewer points. Note that the shape of the clusters are maintained, but the clusters are more sparse with each threshold.

- Employment of a classifier to approximate the Jensen-Shannon (JS) divergence between the candidate and target distributions, ensuring the retention of highly aligned samples.

In addition to demonstrating robust performance in object detection tasks, our comprehensive ablation studies validate the efficacy of each proposed component.

2. Related Work

Unsupervised Domain Adaption (UDA) addresses the challenge of reducing performance decline in the target domain when training with source domain data [30, 38]. Efforts are concentrated on alleviating domain discrepancies between the source and target domains, with techniques falling into categories such as style transfer [36, 44], adversarial learning [18, 22], and distribution alignment [31, 32, 46]. Contrary to the usual practice of simultaneous training on both domains, our methodology involves training the model strictly on a meticulously chosen subset of the source domain, utilizing the target domain exclusively for sample selection purposes.

Active Learning entails the strategic selection of training samples from the learner's perspective [6, 43], trying to achieve better performance comparing to passively training with a pre-annotated training set given the same number of samples. The selection process employs various strategies such as uncertainty-based methods [29, 53], diversity-based methods [1, 7], and query-synthesis techniques [2, 16, 45]. In our framework, the focus is to select labeled training re-

sources that represents our target domain the best, which is independent of the status of the learner.

Sythetic Data Generation (SDG) is the process of creating artificial data and labels to mimic real samples [4, 25, 41]. This approach is beneficial when real data is scarce, expensive to annotate, or has biased distributions. Data synthesis in domain adaption combines probabilistic models that incorporate domain knowledge and generative models ([13, 24] to produce data that accurately represents the target domain [49].

Image Similarity Evaluations are mostly used in assessing generative model. The Fréchet inception distance (FID) [55] measures the distance between the distribution of generated images and real images. Maximum mean discrepancy (MMD) is a kernel-based statistical test to compare distributions [21]. We use these metrics to select training samples that most closely represent the target domain.

Graph-based Clustering posits that the decision boundary for sample division should be situated in areas of low density, while points that are related should be connected through paths traversing regions of high density [9, 28]. Based on this principle, similarity propagation techniques are employed to pinpoint samples with high similarities within the adjacency matrix. This concept underpins our method of graph-based pruning.

Training Data Search [52] and neural data server [51] focus on select a subset from the large source set. Our research builds on these methodologies, concentrating on identifying an effective strategy to optimize both representativeness and diversity in our selection process.

Deep Distribution Distance Approximation originates from the advent of Generative Adversarial Networks (GANs) [20], wherein a discriminator is trained to differentiate between real and generated data, thereby estimating the distance between their distributions to guide generator training. This process allows the discriminator's score to serve as an approximation for various statistical divergences and distances, such as the Kullback–Leibler divergence [20], Jensen-Shannon divergence, and Wasserstein distance [3]. In our work, we leverage the discriminator concept to train a classifier that selects samples from the source set which best align with the target set.

3. Method

In this section, we outline our approach for selecting training samples from an annotated source set within a budget. 3.1 details the problem setup; 3.2 covers feature extraction for efficient search. 3.3 discusses using MMD for initial candidate sampling. Finally, 3.4 describes pruning these candidates to meet budget constraints. Figure 3 shows the architecture diagram for our entire pipeline.

3.1. Problem Formulation

The goal here is to select a representative subset D'_s from an annotated source pool of images D_s under budget of images b, such that a competitive detection model can be obtained based on D'_s. To achieve optimal performance, the detection model should be trained on images with similar distribution to our target distribution. The budget is the maximum number of images we can choose form D_s

We follow a two stage approach for this dataset construction process. The first stage is a target specific search where we try to find clusters of images whose distributions align with the target training set. This stage does not take into consideration the budget b but instead tries to maximize the retrieval of images with similar distributions. This is equivalent to maximizing recall from a retrieval point of view.

In the second stage we construct a similarity graph and introduce our cluster density reduction technique. The goal of this stage is to perform pruning to adhere to the budget constraints. In this stage, we try to strike a balance between selecting similar images (through a classifier guidance) but at the same time trying to prune away redundant images which are too similar. The next sections describe in depth both of the stages.

3.2. Feature Extraction and Representation

To encode the images in both D_s and D_t, we employ Contrastive Language–Image Pre-training (CLIP) model using Vision Transformer (ViT), which leverage a transformer-based architecture specifically designed for image processing. ViT divides an image into patches and processes them sequentially, capturing both local and global information. The effectiveness of ViT latents is further enhanced by training on the CLIP dataset, which jointly trains an image encoder (ViT in this case) and a text encoder to align images with their corresponding textual descriptions. The training dataset of CLIP contains 400 million image-text pairs, which can be particularly beneficial for detecting objects in different settings, angles, or lighting conditions. This broader contextual awareness could lead to better detection performance in complex scenes compared to Inception which is trained on 1.2 million ImageNet images with a fixed set of labels and might be less adaptable to contextual variations.

3.3. Clustering and Ranking

Previous research[5, 10, 48] in domain adaptation has empirically and theoretically established that diminishing the discrepancy between source and target domains significantly enhances model performance on target tasks. This correlation is underpinned by the principle that a lower domain discrepancy reduces the model's generalization error when applied to the target domain.

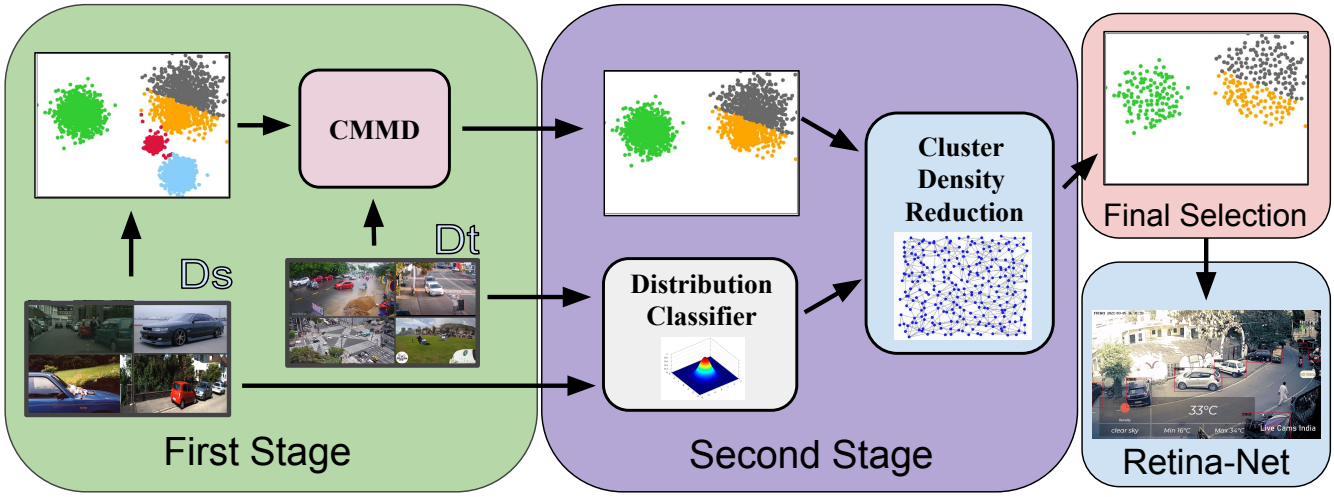

Figure 3. Model architecture: Given an image source D_s, we obtain latents and cluster them into different groups. CMMD is then applied to these clustered latents with the D_t training latents. This generates the filtered candidates $S*$ which are fed into the cluster density reduction framework. A distribution classifier is trained on a subset of D_s and the entire D_t and used to guide the pruning process in the Cluster density reduction operation. This gives the final set of candidates D'_s which satisfy the budget constraint b and are used to train a detection model.

To quantify the discrepancy between each cluster and the target pool, instead of using Fréchet Inception Distance (FID) [23], we compute CMMD score, which is based on the Maximum Mean Discrepancy (MMD) distance and utilizes the rich embeddings from CLIP. MMD operates by embedding distributions into a Reproducing Kernel Hilbert Space (RKHS), a type of function space where the evaluation of functions can be done via inner products. The key idea behind MMD is to compute the difference between the sample means in this RKHS. If the distributions are identical, the difference should be zero; otherwise, the difference quantifies the discrepancy between them. Mathematically, given two distributions P and Q over $\mathbb{R}^d$, the MMD is defined as:

$$\text{dist}^2_{\text{MMD}}(P,Q) = \mathbb{E}_{x,x'\sim P}[k(x,x')] + \mathbb{E}_{y,y'\sim Q}[k(y,y')] - 2\mathbb{E}_{x\sim P, y\sim Q}[k(x,y)] \quad (1)$$

x and x' are independently sampled from distribution P, while y and y' are independently sampled from distribution Q. k is the kernel function which maps the samples into a higher-dimensional space. For two collections of vectors, $X = \{x_1, x_2, \ldots, x_m\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$, drawn from distributions P and Q respectively, the unbiased estimator of $dist^2_{\text{MMD}}(P,Q)$ is provided by the following expression:

$$\hat{\text{dist}}^2_{\text{MMD}}(X,Y) = \frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{\substack{j=1\\ j\neq i}}^{m} k(x_i,x_j) + \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n} k(y_i,y_j) - \frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} k(x_i,y_j) \quad (2)$$

MMD has two main advantages over FID in this case:

1. **No Distribution Assumption**: FID assumes that the feature activations of the Inception network follow a multivariate Gaussian distribution, which may not always hold true for all types of pools of images. MMD, on the other hand, does not rely on such assumptions about the data distribution.
2. **Sample Efficient**: It requires a large sample size (usually above 50k) [11] to calculate FID in order to get a reliable estimation of the $d * d$ covariance matrix, while MMD works well even with a small sample size.

We adopt the target-specific subset search method from [52] to construct the subset S^*. The source images are partitioned into c clusters using k-means clustering on their ViT latents. The clusters are then ranked in ascending order of their CMMD scores, with the lowest score indicating the closest resemblance to D_t. We iteratively select images from the ranked clusters to form the subset D'_s based on their CMMD score, updating S^* with the new cluster if it further lowers the current CMMD score.

This process continues until all clusters have been considered, resulting in a final subset S^* of D_s that minimizes the CMMD score with respect to D_t. This approach ensures that S^* is representative of D_t and is able to train a detection model which can achieve high accuracy.

Algorithm 1: Classifier-guided Cluster Density Reduction

Data: D_s, D_t
Result: D'_s
clustering
$\{c_i\}_c \leftarrow kmeans(D_s)$
initial selection
S^* = target_specific_subset_search($\{c_i\}_c$)
pruning
$S = \{s_{ij}\}_{i,j} = v_i \dot{v}_j^T$ # similarity matrix
$A = \{a_{ij}\}_{i,j} = 1_{\{S_{ij} \geq \tau\}}$ # adjacency matrix
$G = (Edge : \{a_{ij}\}, Vertex : \{v_{ij}\})$ # adjacency graph
$D'_s = \{\}$
for e_{ij} in $\{a_{ij}\}$:
    if size(D'_s) $\geq$ 8000:
        break
    D'_s += $\arg\max_{v_i, v_j} \phi(v)$ # pick the sample with higher classifier score
return D'_s

3.4. Graph Based Pruning

Given the refined dataset S^* from the initial phase, composed of n latent vectors $S^* = \{v_1, v_2, \ldots, v_n\}$ with each $v_i \in \mathbb{R}^d$ derived from the Vision Transformer (ViT) trained on CLIP, our goal is to select a subset $D'_s \subset S^*$ such that $|D'_s| \leq 8000$, optimized for training an object detection model. A similarity matrix $S \in \mathbb{R}^{n \times n}$ is constructed, with each entry $S_{ij} = v_i \cdot v_j^T$ quantifying the similarity between latent vectors v_i and v_j.

Upon establishing a similarity threshold τ, we proceed to an adjacency matrix $A \in \{0,1\}^{n \times n}$ that formulates a bidirectional directed graph $G = (V, E)$, with V representing image vertices and E symbolizing the connecting edges. An edge e_{ij} is forged between vertices i and j if $S_{ij} \geq \tau$, leading to $A_{ij} = 1_{\{S_{ij} \geq \tau\}}$.

To attenuate point density within clusters, for each disjoint subgraph in graph G, we elect either vertex v_i or v_j for inclusion in D'_s and exclude the counterpart. This can be done in several ways. To resolve this selection process, here we describe a classifier based approach that minimize the Jensen-Shannon divergence [39] between selected dataset and the target dataset, which we will be describing in details in section 3.5. We then score the vertices v_i and v_j and pick the vertex with the highest score to be added to D'_s. The algorithm is shown in Algorithm 1. In practice, this can be done efficiently as the similarity matrix can be obtained using matrix multiplication, while using threshold τ can significantly reduce the number of entries in the adjacency matrix.

3.5. Distribution Approximation Classifier

We train a classifier ϕ with the loss function

$$f_\phi = \max_\phi \left(\mathbb{E}_{x \sim D_t(x)}[\log \phi(x)] + \mathbb{E}_{x \sim D_s(x)}[\log(1 - \phi(x)]) \right) \quad (3)$$

Given the target distribution D_t and the source distribution D_s, we can prove that the optimal classifier ϕ^* that maximize f_ϕ exists [20]:

$$\phi^*(x) = \frac{p_{D_t(x)}}{p_{D_t(x)} + p_{D_s}(x)} \quad (4)$$

With the optimal classifier ϕ^*, selecting a subset D'_s from D_s that maximize

$$\max_{D'_s} \mathbb{E}_{x \sim D'_s}[\phi^*(x)] \quad (5)$$

is equivalent to selecting a subset D'_s that minimize the JS divergence between D'_s and D_t [20].

Intuitively, optimized with eq 3, this classifier is trained to distinguish between target samples and source samples (binary classification problem with Binary Cross-Entropy (BCE) loss), by identifying important global features that are specific to the target set - such as camera viewpoint (e.g., surveillance vs. vehicle-mounted cameras), object density (e.g., the number of cars on a road), and etc - which are crucial for the object detection task. The trained classifier is then used to pick samples that share the most similarities from the target set, enabling it to fetch the most aligned training samples.

Combined with the idea of keeping the diversity in training set, the trained classifier is employed to score each vertex v_i and v_j in the similarity graph, with the score reflecting the vertex's similarity to the global attributes of D'_t. The vertex with the highest score is chosen for inclusion in the subset D'_s, effectively maximizing eq 5. This methodical selection process guarantees that D'_s is augmented with images that are representative of the target dataset's distribution and possess key global characteristics relevant to the object detection task.

By focusing on global image attributes, this approach addresses potential domain shifts between the source and target datasets, ensuring that the selected subset D'_s is more aligned with the target domain. This alignment is crucial for improving the generalization ability of the object detection model, as it reduces the risk of overfitting to source-specific features that may not be present in the target domain.

The process supersedes random selection by maintaining the integrity of the original data distribution in D'_s while concurrently expurgating superfluous images that do not

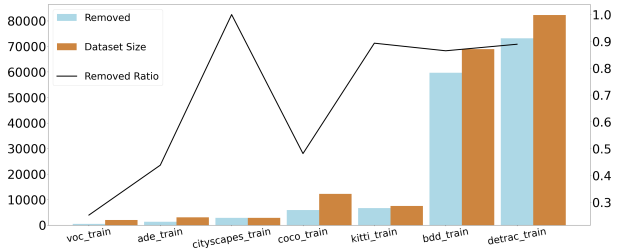

Figure 4. This plot shows the comparison between the total data size vs the amount of data kept after first stage filtering in each of the data sets that comprise the source pool D_s. The bar plots show these numbers while the line plot shows the ratio of images removed vs total number of images in each of the constituent datasets. Cityscapes has the maximum ratio of images removed while voc has the least. In absolute numbers detrac has the most number of images removed among all of the datasets.

contribute to the model's discriminative capacity. Consequently, this approach preserves the intrinsic heterogeneity of the dataset, ascertaining that D'_s is both comprehensive and emblematic of the broader data corpus. Such methodical curation is instrumental in bolstering the object detector's accuracy and reliability. We showed in fig 2 our pruning method on clusters of two-dimensional vectors to illustrate the effect of the pruning. By adjusting the threshold, it changes the sparsity of the clusters, but still maintain the overall shape of the clusters. In section 4 we also demonstrate ablation study on how the classifier outperforms random selection. It preserves the global shape of the clusters while significantly reducing density in each clusters.

4. Experiment

For all our experiments, we follow the dataset introduced in the second DataCV challenge and follow their settings and detection model.

4.1. Experimental Settings

4.1.1 Source and Target Datasets

- **Source Datasets**: Comprise datasets from seven existing sources: ADE [56], BDD [54], Cityscapes [12] , COCO [27], VOC [17], Detrac [50], and KITTI [19]. In total, the collection contains 176,491 images whose labels are related to vehicles
- **Target Datasets**: We use Region100 benchmark as our target dataset.It consists of footage captured by 100 static cameras from various regions in the real world. For videos from each different region, the first 70% (15368 images) is used for model training, while the remaining 30% is designated for validation (2134 images) and testing (4368 images).

Table 1. Object detection mAP results. CDR refers to Cluster Density Reduction

Method	mAP
Random	13.6
Random 8k coco	19.7
SnP with random pruning	18.0
CDR with random pruning	21.7
CCDR	**22.2**

4.1.2 Detection Model

In our experiments, we use RetinaNet model with hyperparameters fixed required by this challenge. The implementation can be accessed in the public repository [2]

4.1.3 Evaluation Metric

We use Mean Average Precision(mAP) as the evaluation metric in our experiments.

4.1.4 Experiment details

Baselines We run several baselines that set benchmarks to test our approach. All the baselines are provided in the table 1. All experiments are conducted on an IBM POWER9 CPUs@3.8GHz, 200GB RAM and NVIDIA A6000 GPU. We run experiments to test the first as well as second stages under different settings. The experiments include random selection baselines, benchmarking with a modified SnP version(random pruning), Cluster Density Reduction with random pruning and Cluster Density Reduction with Classifier Guidance.

We experimented with two feature extraction models to obtain the latents corresponding to the images. For image feature extraction, we use IncepetionV3 [47] pretrained on ImageNet [14] as well as ViT-L/14@336px trained on Clip [42]. For the first stage filtering we use ViT-L/14@336px model to generate embeddings and Gaussian Radial basis function(RBF) kernel $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$ for calculating MMD.

For the second stage cluster reduction we use InceptionV3 latents to calculate the similarity matrix S. We empirically found that this setting works better than using the ViT embeddings for both settings. This could be because the addition of Inception brings more diversity to the retrieval process by incorporating features that ViT latents lack.

4.2. Main Results

We provide a simple random baseline that selects n images randomly to satisfy the budget constraints b. This is

[2] https://github.com/yorkeyao/DataCV2024/tree/main/task_model

Figure 5. Qualitative Analysis of the distribution classifier. The first row shows variety of images that are visually similar to the target dataset. The second row shows visually dissimilar images. Note that even though there are cars in the second row images, they do not look alike as the target dataset

a very weak baseline as evident from Table 1. Next we randomly selected n images from each of the individual data sources that comprise D_s. We found that COCO surprisingly provides a very strong baseline here. We attribute this to it's versatility and the diverse scene variation and backgrounds that help to train a robust model. Additionally to explain this we refer to Figure 4, which plots the ratio as well as number of images removed vs total number of images present in each dataset after the first stage using CCDR. As evident from the graph COCO is among the lowest datasets in terms of images filtered out from the first stage. This combined with its larger size explains why COCO random sample is a strong baseline. A very strong baseline is the SnP[52]. However since we are not working with IDs here in the D_s, the pruning stage is modified to randomly select images post their first search stage. We also evaluate our cluster density reduction approach under two settings. In the first setting, we select 8k from the first stage by randomly picking one image from the pair of similar images. We then change the pruning method by adding guidance from our classifier. We observe that classifier does indeed provide a significant gain.

4.2.1 Distribution Classifier ϕ

For training simplicity, we train the classifier ϕ on the extracted feature embedding x instead of the raw input image. We set the classifier to be a 2 layer MLP (hidden_dim=200) with dropout and batchnorm1d.

Training Set Choices

In our theoretical framework, the training of ϕ involves labeling all samples from the source set as 0 and all samples from the target set as 1. However, practical implementation revealed that the classifier found it too straightforward to distinguish between the source and target sets, primarily because it couldn't effectively propagate gradients through the source images, as opposed to traditional GAN training [20]. This resulted in nearly perfect scoring on all source samples, deteriorating performance of the selection.

Table 2. Ablation on training choices of ϕ.

Training Choices	mAP
random picked negative	21.5
cityscape + kitti as negative	**22.2**
cityscape + kitti + detrac as negative	21.9
coco as negative	21.3

To address this issue, we opted to select a subset of samples from the source set as negative samples for training the classifier. Ablation studies, summarized in Table 2, demonstrate the method's robustness to the choice of the training subset. This approach not only mitigates the classifier's tendency to overfit to the source set but also ensures effective generalization and robustness across different training subsets. Note that only using COCO as negative samples decreased the performance by a few points, we argue that in this case the classifier learnt specific pattern to exclude COCO dataset, which can hurt the performance as COCO is one of the most representative datasets in the training sources, as illustrated in fig 4.

We randomly sampled 10% of the selected source samples and included all images with $\text{region\_id} > 90$ (approximately 10%) from the target set as the validation set.

Qualitative Analysis We also did a qualitative study 5 of the classifier's performance on the source dataset. We let the classifier predict scores for different images in the data

Table 3. Ablation on clustering methods

Clustering method	mAP
K-means	**22.2**
HDBSCAN with Euclidean	21.5
HDBSCAN with manhattan	21.9
HDBSCAN with cosine	21.7
Agglomerative with Euclidean	21.5
Agglomerative with manhattan	21.6
Agglomerative with cosine	21.3

source. We then rank these images on the classifier scores and sample some images from the top as well as from the bottom of the list. As seen from the figure 5, the top rows are the most similar images and a visual inspection shows that these images are indeed very similar to the images in the region100 training set. Conversely, the images in the bottom row are dissimilar to the region100 training set.

4.2.2 Ablation Study

We conduct a series of comparative experiments to evaluate the performance of CCDR using various clustering techniques in Stage 1. Our experiments include applying K-means, HDBSCAN[37] with different distance metrics (Euclidean, Manhattan, Cosine), and Agglomerative Clustering[40] with these same metrics. K-means clustering is executed using CLIP embeddings with a dimensionality of 768. For HDBSCAN and Agglomerative Clustering, we first reduce the CLIP embeddings to a dimensionality of 10 using UMAP, followed by clustering. The number of clusters is set to 75 for both K-means and Agglomerative Clustering. In the case of HDBSCAN, we specify a minimum cluster size of 500, resulting in a total of 50 clusters. As demonstrated in Table 3, K-means achieves superior mean Average Precision (mAP) scores compared to HDBSCAN and Agglomerative Clustering. This suggests that K-means is relatively resilient to the curse of dimensionality in this context. Conversely, dimensionality reduction, while mitigating the curse of dimensionality, appears to compromise the integrity of the embeddings. This reduction in feature space intricacy leads to diminished mAP scores for HDBSCAN and Agglomerative Clustering. Although applying these methods could potentially enhance performance, their computational inefficiency in handling large, high-dimensional datasets undermines the efficiency gained from processing a smaller, representative subset. Consequently, our methodology for Stage 1 prioritizes the use of K-means.

Further we study the impact of changing cluster size in the first stage on the performance of our algorithm. Table 4 shows the results of this ablation study. We varied the cluster size between the ranges 50-100 and sampled points from this range. The results show that cluster size of 75 is the most optimal number for this dataset. Decreasing it below or increasing it above 75 led to poorer performances. For all of our experiments we then set it to the optimal size obtained from this study.

Table 4. Ablation on cluster size.

Cluster number	mAP
50	19.8
70	20.1
75	**22.2**
80	21.9
100	19.6

5. Conclusion

In this paper, we described our two-stage approach- Classifier Guided Cluster Density Reduction (CCDR) for the CVPR 2024 VDU Challenge. In the first stage we employ CMMD with ViT latents to filter out unrelated clusters and select images that are similar in distribution to the target dataset. In the second stage we proposed a classifier guided graph pruning approach to reduce the densities of the selected clusters to adhere to a fixed budget, thus removing redundant images. Our model achieved highly competitive performance in the competition as well as demonstrates superior performance over several baselines and methods in this area.

References

[1] Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 137–153. Springer, 2020. 2

[2] Ibrahim Alabdulmohsin, Xin Gao, and Xiangliang Zhang. Efficient active learning of halfspaces via query synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015. 2

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. 3

[4] André Bauer, Simon Trapp, Michael Stenger, Robert Leppich, Samuel Kounev, Mark Leznik, Kyle Chard, and Ian Foster. Comprehensive exploration of synthetic data generation: A survey. *arXiv preprint arXiv:2401.02524*, 2024. 3

[5] Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, Alex Kulesza, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010. 3

[6] Jun Bi, Xiaqing Li, Qi Guo, Rui Zhang, Yuanbo Wen, Xing Hu, Zidong Du, Xinkai Song, Yifan Hao, and Yunji Chen. Balto: fast tensor program optimization with diversity-based

active learning. In *The Eleventh International Conference on Learning Representations*, 2022. 2

[7] Felix Buchert, Nassir Navab, and Seong Tae Kim. Toward label-efficient neural network training: Diversity-based sampling in semi-supervised active learning. *IEEE Access*, 11: 5193–5205, 2023. 2

[8] Cheng Chang, Himanshu Rai, Satya Krishna Gorti, Junwei Ma, Chundi Liu, Guangwei Yu, and Maksims Volkovs. Semi-supervised exploration in image retrieval. 2019. 1

[9] Olivier Chapelle, Jason Weston, and Bernhard Schölkopf. Cluster kernels for semi-supervised learning. *Advances in neural information processing systems*, 15, 2002. 3

[10] Chao Chen, Zhihong Chen, Boyuan Jiang, and Xinyu Jin. Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 3296–3303, 2019. 3

[11] Min Jin Chong and David Forsyth. Effectively unbiased fid and inception score and where to find them. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6070–6079, 2020. 4

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 6

[13] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018. 3

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009. 6

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[16] Adrian Englhardt and Klemens Böhm. Exploring the unknown–query synthesis in one-class active learning. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pages 145–153. SIAM, 2020. 2

[17] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 6

[18] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016. 2

[19] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. In *The International Journal of Robotics Research*, pages 1231–1237. SAGE Publications Sage UK: London, England, 2013. 6

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 3, 5, 7

[21] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 3

[22] Gewen He, Xiaofeng Liu, Fangfang Fan, and Jane You. Classification-aware semi-supervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 964–965, 2020. 2

[23] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 4

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[25] Yongjun Hong, Uiwon Hwang, Jaeyoon Yoo, and Sungroh Yoon. How generative adversarial networks and their variants work: An overview. *ACM Computing Surveys (CSUR)*, 52(1):1–43, 2019. 3

[26] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. *arXiv preprint arXiv:2401.09603*, 2023. 2

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. *European Conference on Computer Vision*, pages 740–755, 2014. 6

[28] Chundi Liu, Guangwei Yu, Maksims Volkovs, Cheng Chang, Himanshu Rai, Junwei Ma, and Satya Krishna Gorti. Guided similarity separation for image retrieval. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[29] Shang Liu and Xiaocheng Li. Understanding uncertainty sampling. *arXiv preprint arXiv:2307.02719*, 2023. 2

[30] Xiaofeng Liu, Chaehwa Yoo, Fangxu Xing, Hyejin Oh, Georges El Fakhri, Je-Won Kang, Jonghye Woo, et al. Deep unsupervised domain adaptation: A review of recent advances and perspectives. *APSIPA Transactions on Signal and Information Processing*, 11(1), 2022. 2

[31] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer joint matching for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1410–1417, 2014. 2

[32] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. *Advances in neural information processing systems*, 29, 2016. 2

[33] Yichao Lu, Cheng Chang, Himanshu Rai, Guangwei Yu, and Maksims Volkovs. Learning effective visual relationship detector on 1 gpu. *arXiv preprint arXiv:1912.06185*, 2019. 1

[34] Yichao Lu, Cheng Chang, Himanshu Rai, Guangwei Yu, and Maksims Volkovs. Multi-view scene graph generation in videos. *International Challenge on Activity Recognition (ActivityNet) CVPR 2021 Workshop*, 2021.

[35] Yichao Lu, Himanshu Rai, Jason Chang, Boris Knyazev, Guangwei Yu, Shashank Shekhar, Graham W. Taylor, and Maksims Volkovs. Context-aware scene graph generation with seq2seq transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15931–15941, 2021. 1

[36] Robert A Marsden, Felix Wiewel, Mario Döbler, Yang Yang, and Bin Yang. Continual unsupervised domain adaptation for semantic segmentation using a class-specific transfer. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022. 2

[37] Leland McInnes, John Healy, Steve Astels, et al. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205, 2017. 8

[38] Akshay Mehra, Bhavya Kailkhura, Pin-Yu Chen, and Jihun Hamm. Understanding the limits of unsupervised domain adaptation via data poisoning. *Advances in Neural Information Processing Systems*, 34:17347–17359, 2021. 2

[39] ML Menéndez, JA Pardo, L Pardo, and MC Pardo. The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318, 1997. 5

[40] Fionn Murtagh and Pierre Legendre. Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? *Journal of classification*, 31:274–295, 2014. 8

[41] Sergey I Nikolenko. *Synthetic data for deep learning*. Springer, 2021. 3

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2, 6

[43] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021. 2

[44] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8503–8512, 2018. 2

[45] Raphael Schumann and Ines Rehbein. Active learning via membership query synthesis for semi-supervised sentence classification. In *Proceedings of the 23rd conference on computational natural language learning (CoNLL)*, pages 472–481, 2019. 2

[46] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 2

[47] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 6

[48] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 3

[49] Kang Wang, Rui Zhao, and Qiang Ji. A hierarchical generative model for eye image synthesis and eye gaze estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 440–448, 2018. 3

[50] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, and Siwei Lyu. Ua-detrac: A new benchmark and protocol for multi-object detection and tracking. In *Computer Vision–ECCV 2016 Workshops*, pages 525–538. Springer, 2015. 6

[51] Xi Yan, David Acuna, and Sanja Fidler. Neural data server: A large-scale search engine for transfer learning data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3893–3902, 2020. 3

[52] Yue Yao, Tom Gedeon, and Liang Zheng. Large-scale training data search for object re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15568–15578, 2023. 2, 3, 4, 7

[53] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 93–102, 2019. 2

[54] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2020. 6

[55] Yu Yu, Weibin Zhang, and Yun Deng. Frechet inception distance (fid) for evaluating gans. *China University of Mining Technology Beijing Graduate School*, 2021. 3

[56] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 633–641, 2017. 6