deepPIC: Deep Perceptual Image Clustering For Identifying Bias In Vision Datasets

Nikita Jaipuria*, Katherine Stevo†, Xianling Zhang*, Meghana L. Gaopande*
Ian Calle Garcia*, Jinesh Jain*, Vidya N. Murali*
*Ford Greenfield Labs, Palo Alto †Georgia Institute of Technology
{njaipuri, xzhan258, mgaopand, icalle, jjain1, vnariyam}@ford.com, katiestevo1@gmail.com

Abstract

*Dataset bias in manually collected datasets is a known problem in computer vision. In safety-critical applications such as autonomous driving, these biases can lead to catastrophic errors from models trained on such datasets, jeopardizing the safety of users and their surroundings. Being able to unpuzzle the bias in a given dataset, and across datasets, is an essential tool for building safe and responsible AI. In this paper, we present **deepPIC**: **deep P**erceptual **I**mage **C**lustering, a novel hierarchical clustering pipeline that leverages deep perceptual features to visualize and understand bias in unstructured and unlabeled datasets. It does so by effectively highlighting nuanced subcategories of information embedded within the data (such as multiple but repetitive shadow types) that typically are hard and/or expensive to annotate. Through experiments on a variety of image datasets, both open-source and internal, we demonstrate the effectiveness of deepPIC in (i) singling out errors in metadata from open-source datasets such as BDD100K; (ii) automatic nuanced metadata annotation; (iii) mining for edge cases; (iv) visualizing inherent bias both within and across multiple datasets; and (v) capturing synthetic data limitations; thus highlighting the wide variety of applications this pipeline can be applied to. All clustering results included here have been uploaded with image thumbnails on our project website - `https://alchemz.github.io/unpuzzle_dataset_bias/`. We recommend zooming in for best impact.*

1. Motivation

Bias is an inherent and unavoidable property of manually collected datasets. One of the most common forms in which bias shows up in manually collected datasets is a skewed representation of certain elements and/or geo-locations that are easier to collect data for. For example, one of the largest and most diverse datasets, BDD100K [43], has only 0.2% of its annotated objects as motorcycles with a 238:1 ratio between the total number of car and motorcycle instances. For an object detection algorithm, such biases can skew the results towards the prominent object classes such as cars, resulting in low class-wise accuracy and poor generalization performance in the real-world [7, 34]. In this paper, we focus exclusively on identifying bias in vision data pertaining to safety-critical automotive applications, thus laying the first step towards mitigating bias.

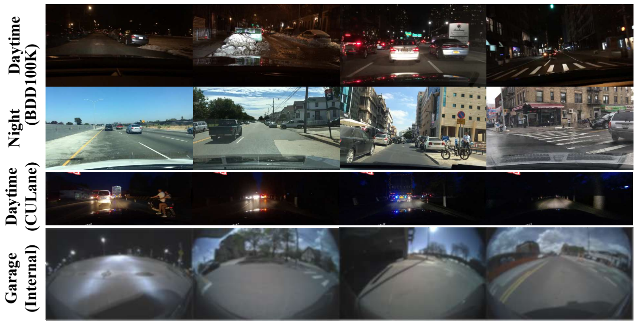


Figure 1. Automatic identification of images with incorrect metadata (left) in both open-source and internal datasets using deepPIC.

Prior research aims at mitigating bias by improving neural network architectures [12, 35–37], training procedures [2, 4, 7, 42] and augmenting datasets [14, 18, 30]. To the best of our knowledge, there is no reliable method available to effectively visualize the amount and type of bias in image datasets. Such a method would be highly instrumental in *targeted bias mitigation* through purposeful data collection/augmentation as opposed to relying on the completion of at least one full model development and testing cycle for the identification of failure modes from existing data gaps. In the case of image datasets, estimating bias on an image pair level is analogous to measuring perceptual similarity. Humans are effortlessly adept at gauging perceptual similarity between two images. However, there isn't a concrete mathematical understanding available of the underlying mechanisms used by humans for this task. Recently, Zhang et al. [45] showed that deep perceptual similarity metrics, relying on image features from neural net-

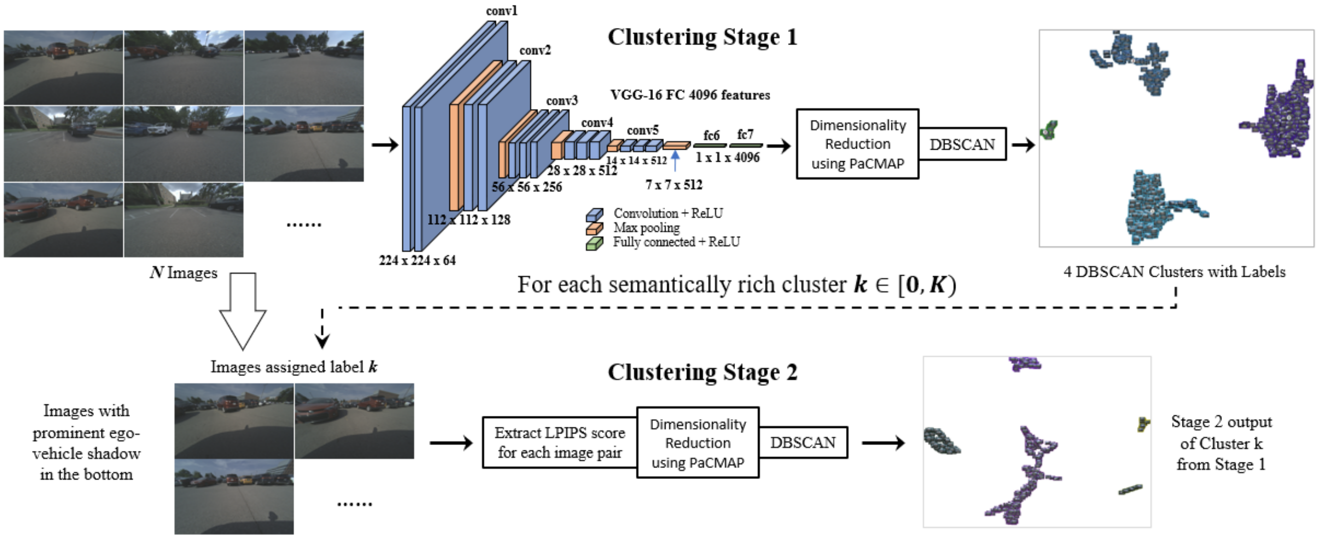

Clustering Stage 1
conv1
conv2
conv3
conv4
conv5
fc6
fc7
VGG-16 FC 4096 features
224 x 224 x 64
112 x 112 x 128
56 x 56 x 256
28 x 28 x 512
14 x 14 x 512
7 x 7 x 512
1 x 1 x 4096
Convolution + ReLU
Max pooling
Fully connected + ReLU
Dimensionality Reduction using PaCMAP
DBSCAN
4 DBSCAN Clusters with Labels
N Images
......
For each semantically rich cluster $k \in [0, K)$
Clustering Stage 2
Images assigned label k
Images with prominent ego-vehicle shadow in the bottom
Extract LPIPS score for each image pair
Dimensionality Reduction using PaCMAP
DBSCAN
Stage 2 output of Cluster k from Stage 1

Figure 2. Schematic of the hierarchical clustering pipeline, deepPIC

works such as SqueezeNet [16], AlexNet [20] and VGG-16 [33] trained on a variety of vision tasks, significantly outperform classical metrics, such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [41], in effectively measuring human perceptual similarity.

Taking inspiration from their findings, this paper presents a novel two-stage hierarchical clustering pipeline called **deepPIC** i.e. **deep** **P**erceptual **I**mage **C**lustering, that leverages deep perceptual features like ImageNet [31] trained VGG-16 [33] activations and Learned Perceptual Image Patch Similarity (LPIPS) [45] to segregate unlabeled datasets into semantically meaningful clusters, with distinct inter- and intra-cluster semantic signatures. Typical metadata accompanying open-source datasets, e.g. BDD100K, captures high-level environmental conditions and scene semantics well, such as time of the day (daytime, night, dawn/dusk) and scene type (highway, city street etc.). However, it fails to capture the more nuanced semantic categories such as dominant shadow types and shape within daytime, varying traffic densities within night, gradual changes in image contrast and/or brightness etc. In Section 2 and Section 3.1, through experiments on multiple open-source and internal datasets, it is demonstrated that deepPIC is highly effective in segregating and/or organizing a variety of datasets into salient and nuanced subcategories of information, which would otherwise be too expensive and time-consuming to annotate.

The impact of such a clustering scheme is highlighted by applying it to multiple applications of immense practical use for computer vision and machine learning practitioners such as: (i) nuanced metadata annotation (in Section 3.1) which in turn can aid informative sampling for building effective training and test sets; (ii) automatic identification of incorrect metadata annotations (in Section 3.2); (iii) mining for edge cases (in Section 3.2); (iv) identifying and visualizing inherent dataset bias across different datasets (in Section 3.3); and (v) visualizing domain gap between real and synthetic datasets (in Section 3.4) as a proxy metric for realism and a guiding beacon for improving photorealism of synthetic datasets [13]. Section 4 will discuss key limitations of deepPIC, followed by a conclusion in Section 5.

2. Hierarchical Clustering Using deepPIC

In this section, an overview of the proposed two-stage hierarchical clustering pipeline, deepPIC, is presented. Fig. 2 provides a detailed schematic and Alg. 1 and Alg. 2 provide the pseudocode for the first and second clustering stages.

2.1. Design Considerations

The first stage of high-level semantic clustering is performed in the feature space of the deepest FC 4096 layer of ImageNet pre-trained VGG-16. The specific choice of the ImageNet pre-trained VGG-16 backbone is inspired by its use in the construction of “perceptual losses” in prior work on neural style transfer [11,21], image synthesis [8] and image super-resolution [19]. In all such works, features from multiple intermediate layers are tapped into for the computation of perceptual losses. In contrast, deepPIC intentionally taps into features from the deepest FC 4096 layer only. This is because in the first clustering stage of deepPIC, the goal is to cluster based on *high-level* semantics only; and in any deep convolutional network, features from the deepest layers provide the highest level of abstraction [44].

The second stage of low-level clustering is performed using LPIPS with an AlexNet backbone. LPIPS was introduced by Zhang et al. [45] as a linearly calibrated (on the Berkeley Adobe Perceptual Patch Similarity (BAPPS) dataset) and more effective variant of perceptual similarity as compared to those using non-calibrated backbones. The specific choice of AlexNet versus other networks such as VGG or SqueezeNet follows the recommendation in [45]. It is also important to note that LPIPS taps into features from multiple intermediate layers and not just the deepest layer, as is the case in deepPIC stage 1. Such a setting aligns well with the stage 2 goal of low-level/nuanced semantic clustering.

In both stages, post deep feature extraction, Pairwise Controlled Manifold Approximation (PaCMAP) [40] is leveraged for dimensionality reduction and visualization of the data structure in a two-dimensional map. This is followed by the use of Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [10, 32] for automatically segregating data points into clusters. We chose PaCMAP over other dimensionality reduction methods such as t-Distributed Stochastic Neighbor Embedding (t-SNE) [38] and Uniform Manifold Approximation and Projection (UMAP) [3], since PaCMAP is known to preserve both local and global structures.

2.2. Algorithm Description

In deepPIC stage 1 as described in Alg. 1, for a given set of N images, $\mathbf{I}$, a resizing operation is first performed to satisfy the input layer requirements for ImageNet pre-trained VGG-16 (line 1). This is followed by FC 4096 feature extraction (line 2) to get the feature set $\mathbf{V}$. PaCMAP dimensionality reduction is then performed on $\mathbf{V}$ to obtain a two-dimensional mapping $\mathbf{T^1}$ (refer line 3). Following the recommendations in [40], learning rate is kept fixed at 1 and the number of neighbors is set as 10 if $N < 10,000$ or as $10 + 15(\log_{10} N - 4)$ if $N > 10,000$. The PaCMAP step is followed by the use of DBSCAN to obtain cluster labels $\mathbf{C^1}$ (the superscript 1 denotes stage 1) for the visualized data structure in $\mathbf{T^1}$ based on the Euclidean distance between clusters (line 4). More specifically, we use the `sklearn.cluster.DBSCAN` [27] implementation with `epsilon` = 0.05 and `min_samples` = 15 chosen empirically based on two main criterion: (i) semantically meaningful; and (ii) visually distinct cluster segregation.

Algorithm 1 deepPIC Stage 1

```
Input: A set of $N$ images, $\mathbf{I} : \{I_i \,\forall\, i \in [0, N)\}$
Output: JSON mapping each image to its perceptual space coordinates and assigned cluster
1: $I_i \leftarrow \texttt{resize}(I_i) \,\forall\, i \in [0, N)$          ▷ Resize to 224 × 224
2: $\mathbf{V} : \{V_i \in \mathbb{R}^N \times \mathbb{R}^{4096}\} \leftarrow \texttt{vgg16}(\mathbf{I})$
3: $\mathbf{T^1} \leftarrow \texttt{pacmap}(\mathbf{V})$          ▷ $\mathbf{T^1} \in \mathbb{R}^N \times \mathbb{R}^2$
4: $\mathbf{C^1} : \{\mathbf{C^1_k} \,\forall\, k \in [0, K), C^1_{-1}\} \leftarrow \texttt{dbscan}(\mathbf{T^1})$
```

Fig. 3 shows the deepPIC stage 1 output, $\mathbf{C^1}$, for 5000 images randomly sampled from the BDD100K training set. Note the automatic visual segregation of night and day images into two distinct clusters, $\mathbf{C^1_0}$ and $\mathbf{C^1_1}$ respectively (subscripts 0 and 1 denote the cluster labels), as corroborated by the BDD100K-provided *time of the day* metadata. Sample images from each cluster are shown at the bottom. In developing and testing deepPIC on multiple datasets, it was observed that while $\mathbf{C^1}$ is sufficient for (i) obtaining high-level semantic clustering such as the day-night split in this case; and (ii) for effectively analyzing data structure and diversity in simple datasets; delving deeper into the intra-cluster spread is often beneficial in obtaining more fine-grained semantic segregation. This is where the hierarchical nature of the pipeline comes in.

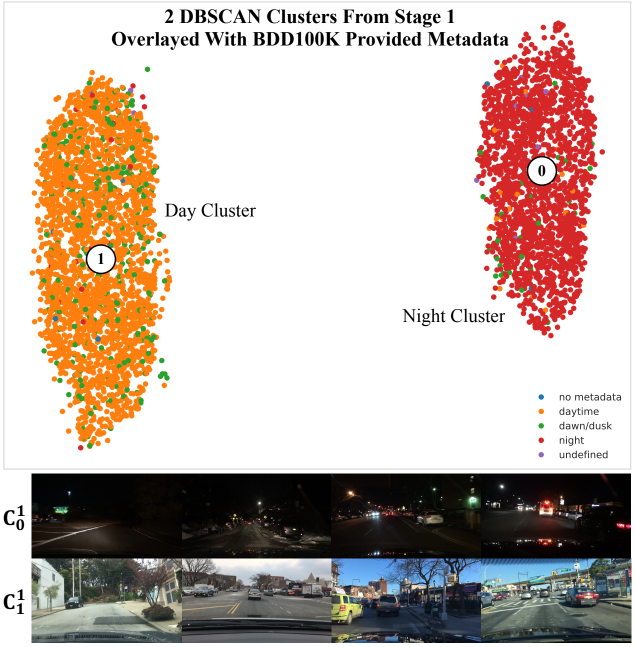


Figure 3. deepPIC stage 1 output ($\mathbf{C^1}$) for 5000 BDD100K images. Note automatic segregation of day and night images into two distinct clusters, as corroborated by the *time of the day* metadata and sample images from each cluster (bottom).

Algorithm 2 deepPIC Stage 2

```
Input: $\mathbf{C^1}$ from Stage 1; Set of $N$ images $\mathbf{I}$
Output: JSON mapping each image to its perceptual space coordinates and assigned cluster
for $k = 0; k < K; k = k + 1$ do          ▷ For all clusters in $\mathbf{C^1}$
    $\mathbf{J_k} \leftarrow \{I_i \,\forall\, i \in [0, N)$ that got clustered in $\mathbf{C^1_k}\}$
    $\mathbf{L_k} \leftarrow \texttt{lpips}(\mathbf{J_k})$ ▷ Extract LPIPS score for each image pair: $\mathbf{L_k} \in \mathbb{R}^{|\mathbf{J_k}|} \times \mathbb{R}^{|\mathbf{J_k}|}$
    $\mathbf{T^2_k} \leftarrow \texttt{pacmap}(\mathbf{L_k})$ ▷ Perform PaCMAP visualization: $\mathbf{T^2_k} \in \mathbb{R}^{|\mathbf{J_k}|} \times \mathbb{R}^2$
    $\mathbf{C^2_k} \leftarrow \texttt{dbscan}(\mathbf{T^2_k})$          ▷ Assign $M_k$ cluster labels
end for
```

For each cluster $\mathbf{C^1_k} \in \mathbf{C^1}$ from stage 1, Alg. 2 performs

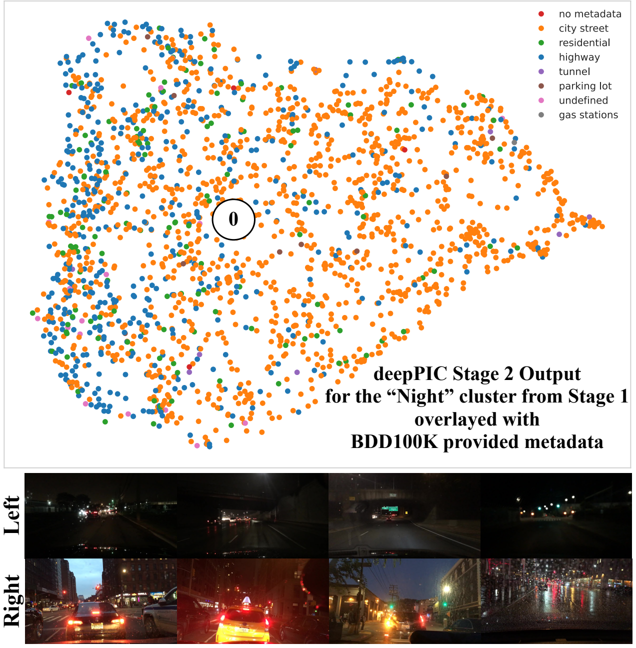


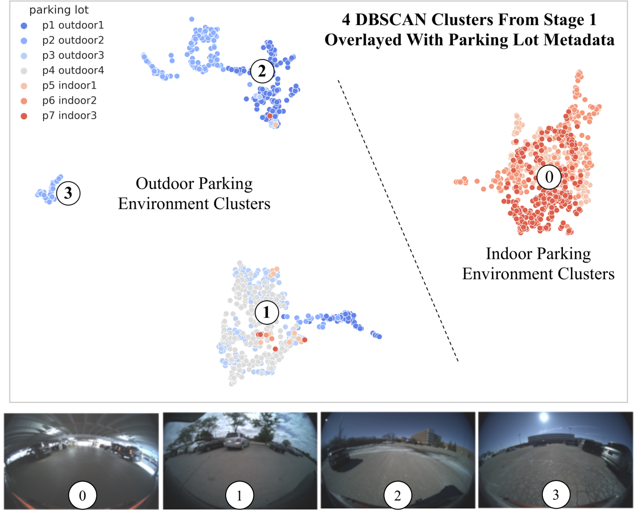
Figure 4. $\mathbf{C}_0^2$, i.e. deepPIC stage 2 output for BDD100K images assigned to $\mathbf{C}_0^1$ (night cluster) in Fig. 3, overlayed with scene metadata. Note the shift from highway/residential to city as we go from left to right. The increase in traffic density from left to right is also shown using sample images (bottom).

a second clustering operation leveraging the LPIPS score. When performing DBSCAN at this stage, we scale the min_samples value of 15 (used in the first stage) to [2, 15] based on the cluster population of $\mathbf{C}_\mathbf{k}^1$. The epsilon value is chosen heuristically between 0.05 and 0.40. Fig. 4 shows $\mathbf{C}_0^2$, i.e. the output of the second clustering stage (denoted by the superscript 2) as applied to night cluster $\mathbf{C}_0^1$ in Fig. 3. Note that while DBSCAN is unable to further segregate the visualized map into distinct sub-clusters, the hierarchical clustering step is still beneficial in organizing all the night images into continuous, semantically evolving trends. For example, the BDD100K *scene* metadata scatter plot (top) in Fig. 4 shows a transition from lower traffic density highway (blue dots) and residential scenes (green dots) in the left, to busy city street scenes (orange points) in the right. Similar insights were obtained from the output of the second clustering stage applied to the daytime cluster $\mathbf{C}_1^1$ in Fig. 3 (included in our project website).

3. Applications

3.1. Pseudo Metadata Annotation

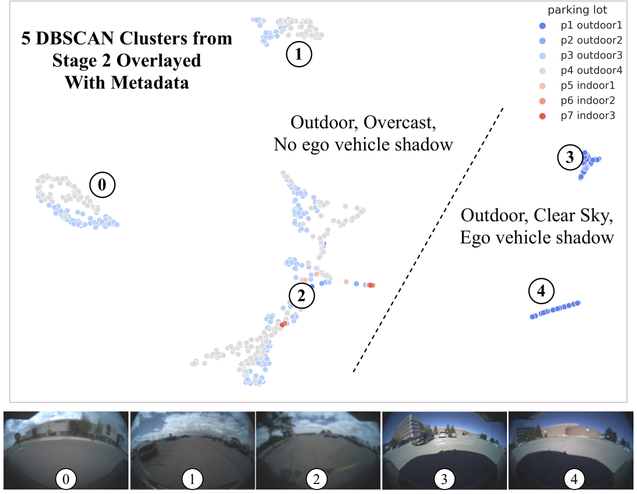
A straightforward application of deepPIC is metadata annotation. Consider an internal parking dataset comprising of 641k images from 7 different parking lots, denoted as p1 - p7 from here on for brevity. p1 - p4 are open parking lots, in close proximity to each other. p5 - p7 are indoor parking garages from geographically similar areas as p1 - p4. A fleet of Sedan vehicles mounted with four cameras was used to collect this data, mostly during daytime. Sample parking images included in the paper have been blurred to ensure no personally identifiable information, such as license plate and pedestrians, are visible. For illustrating pseudo metadata annotation on this dataset, 2k out of 641k, i.e. 0.3% images were sampled uniformly from all 7 parking lots.



Cluster No.	⓪	①	②	③
Environment	Indoor	Outdoor	Outdoor	Outdoor
Time of day	Day/Night	Day	Day	Day
Lighting	Low	Overcast	Bright	Bright
Shadow	Car	Car	Car	Car
Ground	Concrete	Asphalt	Asphalt	Cracked
Lane Color	Yellow	Yellow	Damaged	Damaged

Figure 5. deepPIC stage 1 cluster ($\mathbf{C}^1$) uses 2000 images collected from an internal parking dataset. Note indoor garage data formed $\mathbf{C}_0^1$, which is segregated from those clusters containing images captured from outdoor parking lots ($\mathbf{C}_1^1$, $\mathbf{C}_2^1$, and $\mathbf{C}_3^1$). This split is corroborated by the manually labeled outdoor (cool color dots denoting p1 - p4) versus indoor (warm colored dots denoting p5 - p7) scene metadata.

As described in Section 2, deepPIC, in the first clustering stage, segregates high-level scene semantics. This is followed by a second clustering stage that delves deeper into more nuanced semantics. Fig. 5 shows the deepPIC stage 1 output for the 2k parking images. The three clusters on the left half of the scatter plot ($\mathbf{C}_1^1$, $\mathbf{C}_2^1$, and $\mathbf{C}_3^1$) comprise mainly of outdoor open parking lot images. The remaining cluster, $\mathbf{C}_0^1$, is notably distanced from the rest of the clusters and comprises of images from indoor garages. The different parking structures between indoor garages and open parking lots naturally create different lighting conditions and ground textures, which jointly contribute to the distinct cluster segregation. To annotate this dataset with more nuanced meta information, we present the deepPIC stage 2 clustering results for $\mathbf{C}_1^1$ from Fig. 5 in Fig. 6. Here, all clusters comprise of images from the same domain of outdoor parking lots. However, sub-clusters 3 and 4 are distinct from the rest of the clusters due to the presence of clear skies as opposed to cloudy weather. On taking a closer look, images assigned to these sub-clusters are also distinct due to the presence of a large, hard cast ego vehicle shadow in the image. A more detailed pseudo metadata annotation table is provided on the bottom. The annotation tables in both Fig. 5 and Fig. 6, with the aid of deepPIC, were generated by an engineer within a few minutes. Performing the same task manually would have taken hours.



Cluster No.	⓪	①	②	③	④
Environment	Outdoor	Outdoor	Outdoor	Outdoor	Outdoor
Viewpoint	Left	Rear	Mixed	Right	Left
Weather	Overcast	Overcast	Overcast	Sunny	Sunny
Shadow Type	Occlusion shadow	Occlusion shadow	Occlusion shadow	Ego vehicle shadow	Ego vehicle shadow
Ground	Asphalt	Asphalt	Asphalt	Asphalt, Snow	Asphalt, Snow
Lane Color	Yellow(Y)	Y, Damaged	Y, Damaged	Not Visible	Not Visible
Background	Mall	Mall	Mall	Building	Building

Figure 6. deepPIC stage 2 clustering on $\mathbf{C}_1^1$ creates 5 sub-clusters with distinct metadata segregation: sub-clusters 0, 1 and 2 have overcast skies while 3 and 4 have sunny skies and a strong ego-vehicle shadow.

3.2. Verifying Existing Metadata Annotation And Extracting Corner Cases

In this section, we will demonstrate the effectiveness of deepPIC in: (i) singling out errors in metadata accompanying well-known open-source (BDD100K and CULane [25]) datasets and an internal parking dataset (refer Section 3.1); and (ii) mining for edge cases for vision tasks.

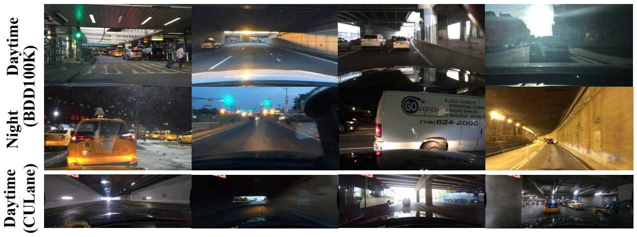


Figure 7. (Top) Edge cases mined using deepPIC from the BDD100K stage 1 *night* cluster, i.e. $\mathbf{C}_0^1$ in Fig. 3 and (Middle) from the day cluster, i.e. $\mathbf{C}_1^1$. (Bottom) Edge cases mined from the CULane stage 1 *day* cluster.

Recall the deepPIC stage 1 clustering output for BDD100K overlayed with time of the day metadata in Fig. 3. Note that $\mathbf{C}_0^1$ (right) comprises mainly of images clustered as *night by appearance*, corroborated by the heavy presence of red dots that denote images labeled night in the BDD100K provided metadata. Similarly, $\mathbf{C}_1^1$ (left) comprises mainly of images clustered as *day by appearance*, again corroborated by the heavy presence of orange dots that denote images labeled daytime in the BDD100K provided metadata. One can notice few red dots in the day cluster and few orange dots in the night cluster. Parsing for these inconsistencies resulted in a total of 26 images that are labeled daytime in the BDD100K provided metadata json but get clustered in the deepPIC *night* cluster; and 27 images that are labeled night in BDD100K metadata but get clustered in the deepPIC *day* cluster. On taking a closer look at these images, we found several that had indeed been incorrectly annotated for time of the day, as shown in Fig. 1. The rest were correctly annotated but are interesting corner/edge cases, as shown in Fig. 7. The top row shows daytime edge cases such as an airport pick up/drop off scene; tunnel/underpass scenes; and strong glare. The middle row shows night edge cases such as a scene with multiple bright yellow cabs on a snow covered street; a late sunset scene with bright glare from the sky and traffic lights; a white van in close proximity covering most of the image; and a well lit tunnel scene.

Repeating a similar analysis on 2000 images sampled from the CULane test set (because metadata is provided only for the test set), we found 43 images that were incorrectly labeled as day in the provided metadata (or more specifically, not labeled as night since there is only a night category in the provided time of the day metadata). On taking a closer look at these 43 images, we again found quite a few that were captured at night but not labeled as night (shown in Fig. 1) and the rest were interesting corner/edge cases such as dark tunnel scenes or indoor parking environments (shown in Fig. 7 bottom row).

For the internal parking data described in Section 3.1, Fig. 5 shows the deepPIC stage 1 output overlayed with metadata annotation that was available for outdoor parking lots (`p1` - `p4`) versus indoor garage (`p5` - `p7`). Note that $\mathbf{C}_1^1$ and $\mathbf{C}_2^1$ mostly comprise of cool color dots which denote images labeled as outdoor parking lots. However, in both clusters, there are some warm color (red) dots. On taking a closer look at these inconsistencies, we found images that had been incorrectly labeled as indoor garage data. Sample images are shown in Fig. 1.

To summarize, deepPIC is effective in automatically identifying incorrect metadata annotations by leveraging inconsistencies between data clusters formed via deep perceptual similarity and manually assigned metadata labels.

3.3. Visualizing Inherent Dataset Bias

Datasets collected for the same task, when collected in silos, tend to display an inherent bias, as highlighted famously by Torralba et al. [34] through their *Name That Dataset* experiment in which a 12-way linear SVM classifier was trained to successfully distinguish between 12 object recognition datasets. For autonomous driving datasets in particular, this inherent bias is mostly a consequence of the fact that it is quite hard to collect data across geographical locations and with diverse environmental conditions.Applying deepPIC stage 1 clustering using Alg. 1 to a randomly sampled mix of 5 popular open-source lane-detection datasets - ApolloScape [15], BDD100K, CULane, Mapillary [22] and TuSimple [1] - is highly effective in confirming this inherent dataset bias (see Fig. 8). Out of the 5 chosen datasets, BDD100K is known to be the most diverse given the large scale, crowd-sourced data collection methodology adopted by its creators. This characteristic is corroborated by the fact that the red dots, representing BDD100K images are intermingled with dots of other colors representing all other datasets. Similarly, given the small size and low diversity in geo-locations, time of the day and weather types in the TuSimple dataset, it stands out, well segregated from all other datasets, as the purple cluster on the right. Overall, the cluster separation shows the inherent bias across public lane datasets.

3.4. Visualizing Sim-to-Real Gap

Given the time and cost intensive nature of real-world data collection coupled with the fast-paced developments in gaming engine based simulation [9, 28], Generative Adversarial Networks (GANs)-based image style transfer [17, 21, 26, 46] and neural rendering [23, 24]; synthetic data augmentation is now widely being leveraged for the creation of richer, more diverse training sets for supervised learning based downstream vision tasks in real-world industrial settings (see Refs. [5, 18] and Tesla's AI day announcement[1]). However, in the absence of reliable metrics for quantifying the sim-to-real gap between generated and real datasets, synthetic data augmentation may end up *hurting* model performance rather than improving it [17]. In this context, another salient application of deepPIC is to use it to visualize the sim-to-real gap, as perceived by deep convolutional neural networks. Fig. 9 highlights one such visualization of a mix (10k images) of real, simulated and sim-to-real GAN translated images (equally split) from an internal parking dataset. Sample images of each type are also shown in the same figure. The simulated images were generated using an in-house Unreal Engine-based simulation tool. The sim-to-real GAN translated images were generated by applying a sim-to-real model based off [17, 21] and trained from scratch on the given real and simulated datasets. Fig. 9 shows $\mathbf{C^1}$, i.e. DBSCAN clusters for the PaCMAP visualization of ImageNet pre-trained VGG-16 features for the full set of 10k images, overlayed with the source of each image (among real, simulated and sim-to-real GAN). 17 clusters are obtained with `epsilon` = 0.05 and `min_samples` = 15. Top highlights include: (i) *sim* (orange) and *real* (blue) data points are non-overlapping indicating a high sim-to-real gap; (ii) *sim-to-real* (green) data points bridge the gap between *real* (blue) and *sim* (orange) data points showcasing a successful domain-translation of the *sim* (orange) images. Such a tool is ground-breaking in finding novel, application-centric ways of identifying the sim-to-real gaps in any dataset.

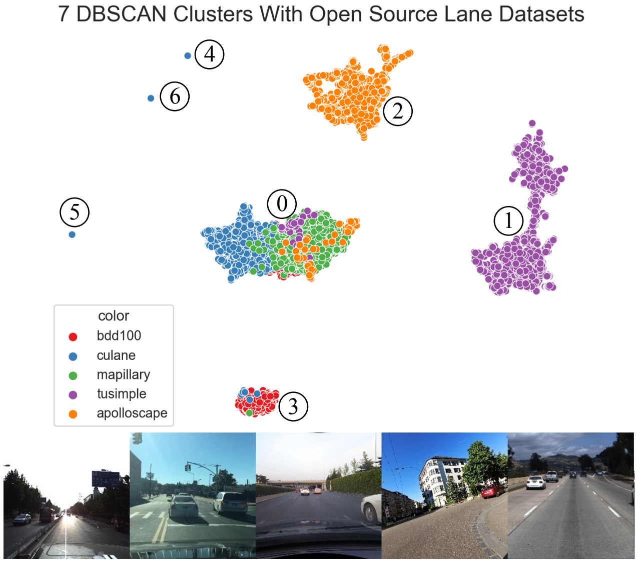


Figure 8. Visualizing inherent dataset bias in open-source lane detection datasets. **(Top)** Stage 1 clustering output from deepPIC applied to 10000 images from 5 different datasets. **(Bottom L-R)** Sample images from ApolloScape, BDD100K, CULane, Mapillary and TuSimple.

4. Discussion

4.1. Limitations

Lack Of Semantic Distinction Across Nearby Clusters: Fig. 10 zooms into the deepPIC stage 1 output for the internal parking dataset described in Section 3.1 to illustrate

[1] https://youtu.be/j0z4FweCy4M?t=5715

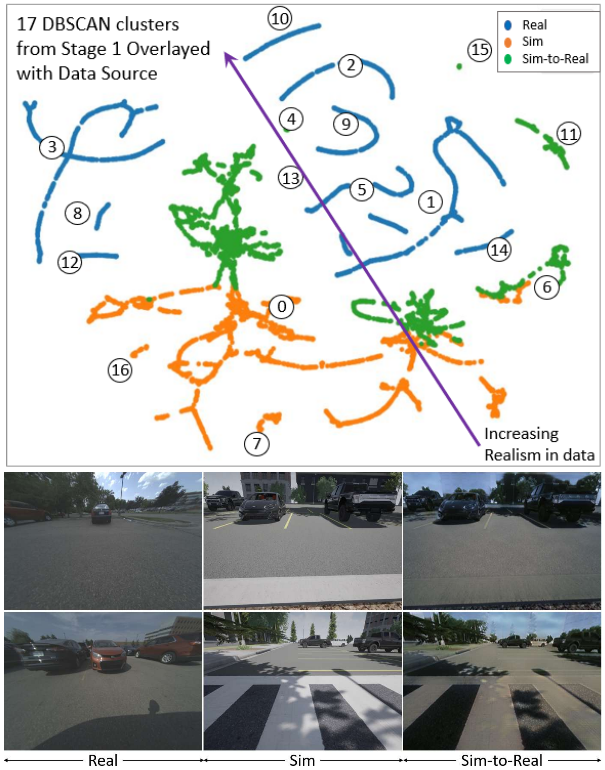


Figure 9. Visualizing sim-to-real gap using $\mathbf{C^1}$, i.e. stage 1 output from deepPIC applied to an equally split mix of 10k real, simulated and sim-to-real GAN translated parking images. The gradual progression of realism between the three sets of data confirms the efficacy of the data augmentation steps. Bottom rows show sample *real, sim* and *sim-to-real* images.

the lack of a meaningful semantic segregation across nearby clusters, particularly where smaller clusters are separated around a large cluster. In this case, $\mathbf{C^1_0}$ and $\mathbf{C^1_2}$ consist of images that are semantically similar to the images in $\mathbf{C^1_1}$ and would perhaps be better placed within $\mathbf{C^1_1}$. Such leaked cluster formations, consisting of a small set of images, can require human-in-the-loop intervention.

LPIPS compute challenges: As described in Section 2, deepPIC Stage 2 clustering leverages the LPIPS score for computing perceptual similarity between pairs of images. This step scales as $\mathcal{O}(N^2)$ where N is the number of images and is a significant compute bottleneck in scaling deepPIC to large-scale datasets. Future work will investigate LPIPS' derivatives that retain its desirable properties in terms of capturing perceptual similarity while also being computationally less expensive.

4.2. Quantitative Analysis

Quantifying clustering effectiveness: Existing metrics such as Rand Index (RI) [29], Adjusted Rand Index (ARI) [39], Normalized Mutual Information (NMI) and Adjusted Mutual Information (AMI) [6], although excellent metrics to capture the effectiveness of any clustering method, require ground truth labels. Thus, they are not applicable as-is to deepPIC, since its focus is on providing pseudo meta-data annotations for datasets which have little/no labels. On the datasets for which some level of meta-data annotations are available as ground truth, we ran additional experiments to quantify the clustering effectiveness of deepPIC. Fig. 3 shows the deepPIC stage 1 output for 5k BDD100K images. It is visually clear that deepPIC does a great job at automatically segregating day and night images into two distinct clusters. Since BDD100K provides time of the day meta-data, we used it as ground truth to provide a quantitative comparison with prior clustering methods. Table 1 shows that deepPIC outperforms all prior methods that cluster in the pixel space.

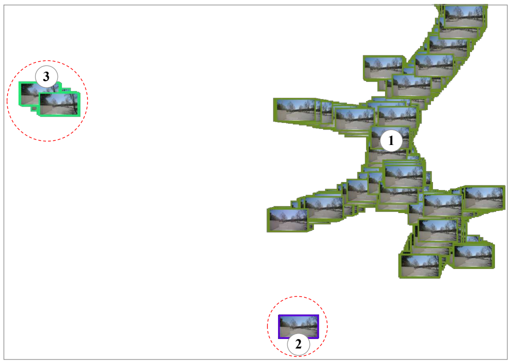


Figure 10. stage 1 output ($\mathbf{C^1}$) from an internal parking lot dataset of 10000 randomly sampled images. Note that the segregation of images shows how smaller leaked clusters can form around a large cluster, that may not be semantically meaningful or significant.

Method	RI	ARI	NMI	AMI
deepPIC Stage 1	**0.90**	**0.80**	**0.74**	**0.74**
PaCMAP† + DBSCAN	0.88	0.76	0.68	0.68
PaCMAP† + K-means	0.88	0.76	0.69	0.69
PCA + DBSCAN	0.44	0.00	0.01	0.01
PCA + K-means	0.89	0.79	0.72	0.72

Table 1. Quantitative comparison with prior clustering methods for BDD100K. Higher values are better. PaCMAP† refers to PaCMAP with the default PCA initialization [40]

A similar comparison was done for the results shown in the Fig. 5 of the deepPIC stage 1 output for 2k images from the internal parking dataset. Here also, deepPIC does a great job at segregating indoor garage images from open parking lot images. For quantifying this result, we leverage internal meta-data annotation for indoor garage versus outdoor parking lots. As shown in Table 2, deepPIC significantly outperforms other clustering methods. The difference between deepPIC and prior methods is greater in this example because of the greater challenge posed by an indoor vs. outdoor segregation task, as opposed to the relatively lower challenge posed by the day vs. night segregation task in the previous example of BDD100K.

Method	RI	ARI	NMI	AMI
deepPIC Stage 1	**0.98**	**0.96**	**0.93**	**0.93**
PaCMAP† + DBSCAN	0.51	0.01	0.01	0.01
PaCMAP† + K-means	0.58	0.16	0.26	0.26
PCA + DBSCAN	0.51	0.00	0.00	0.00
PCA + K-means	0.59	0.19	0.26	0.26

Table 2. Quantitative comparison with prior clustering methods for Parking dataset. Higher values are better. PaCMAP† refers to PaCMAP with the default PCA initialization [40]

It is worthwhile noting that note that such comparisons, even though quite promising in establishing the superiority of deepPIC over other clustering methods, is limited in its ability to quantify the effectiveness of deepPIC in clustering unorganized datasets for downstream applications, such as, pseudo annotations; identifying incorrect annotations; identifying dataset bias; visualizing sim-to-real gap; and so on.

Quantifying Dataset Bias: As shown in Sections 2 and 3, deepPIC is highly effective is visualizing and understanding dataset bias. However, bias quantification remains unaddressed. With the current pipeline, proxy bias quantification metrics can be derived from the number and density of clusters. For example, deepPIC stage 1 on simpler, less diverse datasets, such as the internal object detection dataset described in Section 3.1, results in a large number of sparse and distinct clusters, as shown in Fig. 5 and Fig. 6. In contrast, when applied to a much larger and diverse open-source dataset such as BDD100K, deepPIC results in fewer, denser clusters, with strong semantic evolution within the cluster, as shown in Fig. 3. Thus, a natural next step towards bias quantification would be to quantify such trends to highlight bias.

4.3. Task Specific Bias

For the analysis presented in this paper, ImageNet-trained VGG-16 (deepPIC Stage 1/Alg. 1) and LPIPS with an AlexNet backbone, also trained on ImageNet (deepPIC Stage 2/Alg. 2), are used to derive the deep perceptual features for clustering. As shown in Section 3, both these backbones are highly effective in visualizing underlying semantic and perceptual structure in image datasets. However, a strong case can be made for replacing these generic backbones trained for object detection with task specific backbones. For example, when analysing and curating datasets for the task of lane detection, it would be interesting to compare the output of deepPIC with the VGG-16 and/or AlexNet backbones replaced with a pre-trained lane detection model such as Spatial CNN (SCNN) [25] for the extraction of deep perceptual features. For the lane detection task, this could help highlight task-specific bias such as dominant lane marker types, lane marker condition, ground types etc. Thus, such an analysis could help provide *deeper insights into dataset bias by combining the task model and the dataset itself.*

4.4. deepPIC Informed Sampling

When working with production-scale datasets for automated driving perception tasks, comprising of millions of images, the kitchen sink approach of annotating and learning from all available data is not only expensive, but also highly susceptible to overfitting. A naive alternate is to randomly sample smaller subsets. However, random sampling can result in loss of under-represented scenarios and edge cases. Future work will investigate the use of deepPIC informed sampling as an alternate to random sampling in such scenarios for effective training and test set curation.

5. Conclusion

Dataset bias can creep into an AI/ML pipeline at various development stages, in many cases, implicitly. Being able to unpuzzle that bias in a given dataset and across datasets is a powerful tool. With the proposed hierarchical clustering pipeline, deepPIC, this paper presents a novel data-centric way to leverage deep perceptual features and similarity metrics such as ImageNet trained VGG-16 activations and LPIPS, to understand inter- and intra-cluster relationships in unstructured image datasets. Rich and insightful visualizations are obtained using PaCMAP. This method, showcased on diverse vision datasets, works very well in exposing pseudo metadata annotations which are confirmed by a human-in-the-loop. Nuanced annotations can thus be obtained at a much lower overall cost. The tool has been designed to identify dataset bias, both natural and due to human induced errors. While not meant for directly building machine learning models or deploying models in customer facing applications, deepPIC is designed to be used offline as a visual template for inspecting and interpreting the analysis of data used in development of machine learning models. In fact, the tool can be used to detect biases and erroneously represented data that can be harmful/sensitive for end-user applications. This, along with other wide variety of detailed applications, are presented to emphasize the broad spectrum of benefits that can be derived. We are excited about this work laying the foundations for further techniques for understanding and eventually mitigating dataset bias, ultimately building *responsible AI.*

6. Acknowledgements

We would like to thank Dimitar Filev for insightful discussions and feedback; and Raju Nallapa for constant support and motivation. We are also indebted to Marcos Paul Gerardo Castro, Kunle Olutomilayo and Shubham Shrivastava for reviewing our submission draft and providing constructive criticism.

References

[1] TuSimple Lane Detection Challenge. https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane_detection, 2017. [Online]. 6

[2] Ehsan Adeli, Qingyu Zhao, Adolf Pfefferbaum, Edith V. Sullivan, Li Fei-Fei, Juan Carlos Niebles, and Kilian M. Pohl. Representation learning with statistical independence to mitigate bias. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2512–2522, 2021. 1

[3] Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using umap. *Nature biotechnology*, 37(1):38–44, 2019. 3

[4] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29:4349–4357, 2016. 1

[5] Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, Sergey Levine, and Vincent Vanhoucke. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. *CoRR*, abs/1709.07857, 2017. 6

[6] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999. 7

[7] Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73, 2018. 1

[8] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *Advances in neural information processing systems*, 29, 2016. 2

[9] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 6

[10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996. 3

[11] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 2

[12] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2066–2073. IEEE, 2012. 1

[13] 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. 2

[14] Alexandru Gurghian, Tejaswi Koduri, Smita V Bailur, Kyle J Carey, and Vidya N Murali. Deeplanes: End-to-end lane position estimation using deep neural networksa. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 38–45, 2016. 1

[15] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2702–2719, 2019. 6

[16] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 2

[17] Nikita Jaipuria, Shubh Gupta, Praveen Narayanan, and Vidya N Murali. On the role of receptive field in unsupervised sim-to-real image translation. *arXiv preprint arXiv:2001.09257*, 2020. 6

[18] Nikita Jaipuria, Xianling Zhang, Rohan Bhasin, Mayar Arafa, Punarjay Chakravarty, Shubham Shrivastava, Sagar Manglani, and Vidya N Murali. Deflating dataset bias using synthetic data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 772–773, 2020. 1, 6

[19] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 2

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 2

[21] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pages 700–708, 2017. 2, 6

[22] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017. 6

[23] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 6

[24] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2856–2865, 2021. 6

[25] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 5, 8

[26] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, pages 319–345. Springer, 2020. 6

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss,

V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 3

[28] Weichao Qiu and Alan Yuille. Unrealcv: Connecting computer vision to unreal engine. In *European Conference on Computer Vision*, pages 909–916. Springer, 2016. 6

[29] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971. 7

[30] Kanishka Rao, Chris Harris, Alex Irpan, Sergey Levine, Julian Ibarz, and Mohi Khansari. Rl-cyclegan: Reinforcement learning aware simulation-to-real. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11157–11166, 2020. 1

[31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 2

[32] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017. 3

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[34] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pages 1521–1528. IEEE, 2011. 1, 6

[35] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE international conference on computer vision*, pages 4068–4076, 2015. 1

[36] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 1

[37] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 1

[38] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 3

[39] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11:2837–2854, 2010. 7

[40] Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. Understanding how dimension reduction tools work: an empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization. *arXiv preprint arXiv:2012.04456*, 2020. 3, 7, 8

[41] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 2

[42] Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8919–8928, 2020. 1

[43] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020. 1

[44] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 2

[45] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 1, 2, 3

[46] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 6