Demonstration of VCR: A Tabular Data Slicing Approach to Understanding Object Detection Model Performance

Jie Jeff Xu
Georgia Institute of Technology
jxu680@gatech.edu

Saahir Dhanani
Georgia Institute of Technology
saahir@gatech.edu

Jorge Piazentin Ono
Bosch Research North America
jorge.piazentinono@us.bosch.com

Wenbin He
Bosch Research North America
wenbin.he2@us.bosch.com

Liu Ren
Bosch Research North America
liu.ren@us.bosch.com

Kexin Rong
Georgia Institute of Technology
krong@gatech.edu

ABSTRACT

In this demonstration, we present VCR, an automated slice discovery method (SDM) for object detection models that helps practitioners identify and explain specific scenarios in which their models exhibit systematic errors. VCR leverages the capabilities of vision foundation models to generate segment-level visual concepts that serve as interpretable explanation primitives. By integrating these visual concepts with additional image metadata in a tabular format, VCR uses a scalable frequent itemset mining-based technique to identify common patterns associated with model performance. We will demonstrate VCR's capabilities through three usage scenarios. First, users can explore the automatically extracted visual concepts and their associated labels. Second, users can run slice finding on a large object detection dataset and visually inspect the results to discover systematic errors. Finally, users can iteratively refine their slicing results by providing feedback on the granularity of visual concepts and the quality of the generated labels. These scenarios will illustrate how VCR can aid practitioners in discovering non-trivial gaps in their models' performance, providing actionable insights for model improvement.

PVLDB Reference Format:
Jie Jeff Xu, Saahir Dhanani, Jorge Piazentin Ono, Wenbin He, Liu Ren, and Kexin Rong. Demonstration of VCR: A Tabular Data Slicing Approach to Understanding Object Detection Model Performance. PVLDB, 17(12): 4453 - 4456, 2024.
doi:10.14778/3685800.3685898

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/d2i-lab/VCR.

1 INTRODUCTION

As computer vision models continue to advance and are adopted, it is increasingly important that practitioners vet their model's performance not only in the average case but also within specific subsets (or slices) of settings. For example, variations in object recognition accuracy of up to 20% have been observed between images taken in countries with different income levels, due to objects appearing in different contexts [5]. As a result, identifying these problematic data slices can help practitioners improve their training datasets, fine-tune their models, and compare different types of models.

Identifying meaningful data slices in unstructured image datasets is challenging, especially for object detection tasks. While tabular data slices are typically defined by attribute-value pairs (e.g., $age < 20, gender = M$), such explicit attributes are absent in image data. Although existing slice discovery methods (SDMs) can create slices for image classification tasks without explicit attributes, they face two key limitations when applied to object detection.

First, object detection requires a more fine-grained understanding of images at the segment level, as nearby objects may affect bounding box localization and classification more than remote objects. Several prior works [6, 8, 13] use Gaussian mixture models to cluster image-level representations to create slices but do not generalize well to object detection tasks, which require explanations for specific objects rather than entire images. Most similar to our work, Dadvar et al. [4] explain CNN image classification models with segment-level concepts but rely on the models' internal activations to identify these concepts. Second, object detection tasks offer rich metadata information, such as bounding box sizes and locations that are not present in the image classification tasks. For example, ground-truth bounding boxes that are small (size), unusually stretched/compressed in dimensions (aspect ratio), or overly "crowded" with other detection results might negatively impact prediction performance.

To address the above limitations, we introduce VCR, an automated, model-agnostic slice discovery framework that leverages the fine-grained segment-level understanding and rich image metadata to discover human-interpretable slices where object detection models struggle. VCR uses pre-trained vision foundation models to extract visual concepts, such as objects and object parts, and combines them with additional metadata in a unified tabular format. It can then apply a frequent itemset mining-based method to identify problematic data slices. In this demonstration, we will showcase how practitioners may interact with VCR to uncover the various concepts "hidden" in their datasets, identify problematic slices for their models in terms of these concepts and additional metadata, and iteratively refine their slice results by adjusting concept granularities or relabeling the concepts.

Proceedings of the VLDB Endowment, Vol. 17, No. 12 ISSN 2150-8097.
doi:10.14778/3685800.3685898

2 SYSTEM OVERVIEW

Given an input image dataset and model object detection results, VCR processes the data in three main steps: ❶ Concept Discovery, ❷ Tabularization, and ❸ Data Slicing.

2.1 Concept Discovery

Concept discovery aims to identify and group related objects or attributes across images, allowing a collection of these objects to represent a single concept. These visual concepts then form the basis for interpretable explanations. This process is performed in an unsupervised fashion to avoid constraints on a predefined label set.

First, VCR breaks down images into finer-grained representations by using Meta's Segment Anything Model (SAM) [10]. We choose SAM because it can extract segments for various objects and scenes, even for those unseen in training. To incorporate semantic information into image segments, we use MaskCLIP [7], which distills full-image representations to masked ones via masked self-distillation. Next, we derive segment-level CLIP [14] embeddings by averaging MaskCLIP's pixel-level embeddings within each of the segments produced by SAM.

Inspired by ACE [9], we run K-Means clustering over the segment embeddings to group semantically related segments into visual concept clusters such that each cluster represents a single visual concept. For example, a concept of dogs may be formed from the clustering of many segments of dogs across various images. Additionally, VCR leverages the multi-modal nature of the embeddings to attach labels to concept clusters and further enhance interpretability. Given a list of predefined labels, which can be sourced independently from the Internet or provided by the user in an open vocabulary manner, we assign each cluster to the label with the smallest embedding distance.

The quality of VCR depends on the quality of the segmentation and embedding models used. For example, the smallest SAM model (ViT-B) would produce less fine-grained, lower-quality segments. Similarly, some of MaskCLIP's pixel embeddings would be inaccurate and create incoherent concepts. However, with the rapid evolution of foundational models, we expect these models to improve significantly over time.

2.2 Tabularization

Tabularization involves summarizing the interactions between the visual concepts, image metadata, and the object detection model's bounding box predictions into a unified tabular format. This step prepares the data for the subsequent tabular data mining process.

Before identifying the bounding box and concept interactions, VCR performs bounding box matching by pairing the model's bounding box predictions with ground-truth bounding boxes. We specifically employ a greedy matching algorithm that maximizes prediction confidence followed by intersection over union score (IoU), measuring how well pairs of boxes align. This is a necessary step that allows us to identify false positives, false negatives, and misaligned bounding boxes.

We then summarize the interactions between bounding boxes and concepts using an interaction matrix I. If a bounding box pair "m" and an image segment belonging to concept cluster "n" have overlapping area that is more than a threshold, we increment the matrix $I[m][n]$ to count the number of interactions between the concept and bounding boxes that occurred. We also incorporate image metadata to describe the bounding boxes themselves. Specifically, we keep track of the bounding box size, aspect ratio, and the number of overlapping bounding boxes, which we call "crowding." As these interactions are limited in number, VCR utilizes sparse matrices to store the interaction data efficiently.

2.3 Data Slicing

Finally, VCR integrates tabularized bounding box data with object detection error metrics to identify problematic slices through frequent itemset mining, pinpointing areas where the model excels and falters. Object detection models can produce various errors [2], including localization (correct class, incorrect position), classification (correct position, incorrect class), false positives, and false negatives errors. While VCR supports all these error types, our demo focuses on localization errors using the IoU metric, measuring how well predicted bounding boxes align with their ground truth bounding boxes.

VCR employs the Apriori frequent pattern mining technique [1] to identify problematic data slices, represented as "itemsets," by iterating through various slice combinations and calculating their error metrics. Itemsets describe how bounding boxes interact with concepts and their metadata, with concept presences defined as having a value greater than 0 and absences as having a value of 0. For example, the itemset *{person=3, car=0, crowding=[3-5]}* represents the subset of bounding boxes interacting with three segments of a person concept and experiencing a "high" level of crowding with the absence of a car concept. As each itemset represents a specific data subset, VCR also reports metrics that describe the subset size (support count), overall accuracy, and divergence in accuracy. The divergence in accuracy follows that of [12], representing the deviation from mean accuracy. VCR applies additional pruning rules to improve mining efficiency.

3 DEMONSTRATION SCENARIOS

To motivate the use cases, imagine a user working with an object detection model for autonomous driving. The model works with particularly challenging data, featuring diverse settings and unique objects that could be difficult to capture and summarize using traditional segmentation techniques. VCR addresses these challenges through three key scenarios:

1. **Visual Concept Exploration.** The Explorer interface provides an interactive 2D visualization of the automatically generated visual concept clusters and their associated labels. Here, users can analyze the various concepts embedded in their dataset and identify potential relationships with specific classes.
2. **Finding Problematic Data Slices.** The Miner interface allows users to explore and "bookmark" problematic data slices on which an object detection model exhibits poor performance. This allows users to gain insights into the specific patterns that contribute to the model's systematic errors.
3. **Concept Refinement.** VCR allows users to interactively provide feedback to improve slice descriptions by adjusting concept granularity and relabeling existing concepts.

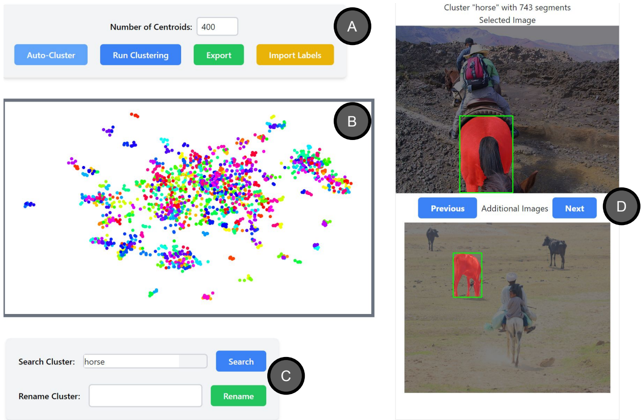


Figure 1: The Explorer page contains the concept embedding overview, cluster modification tools, and concept visualization panel.

During the demo, we will showcase each of these scenarios using one of two datasets: the COCO object detection 2014 validation dataset [11] or a 10k subset of the BDD100k driving dataset [18]. For each dataset, we include detection results from two object detection models provided by the MMDetection library [3]: Faster R-CNN [16] and YOLOv3 [15].

3.1 Visual Concept Cluster Exploration

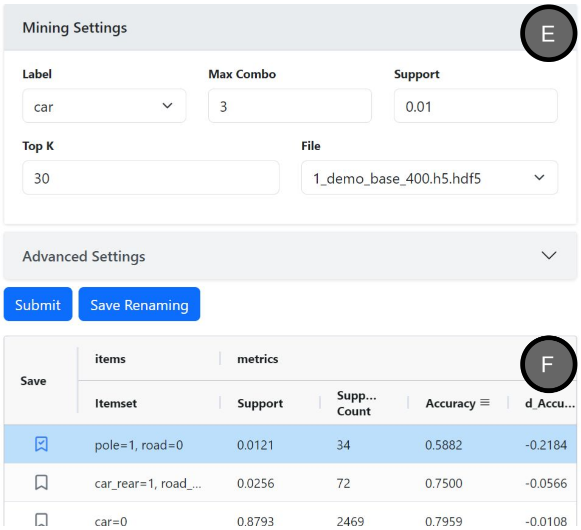
The Explorer page allows users to gain a better understanding of the landscape of their data. By exploring visual concepts, users can obtain an overview of the various objects present in the dataset and identify potential diversity gaps where certain concepts may be overrepresented, underrepresented, or entirely absent. Concept clustering and exploration also serve as a necessary preprocessing step before slice finding.

To start, the user will specify the number of concepts to be formed via K-Means and click the "Run Clustering" button (Figure 1, section A). If the user is uncertain of how many clusters to utilize, VCR offers an "Auto-Cluster" option that automatically chooses the number of clusters using the elbow method based on silhouette scores [17]. The clustering results will be displayed in a plot illustrated in section B, where each point represents an image segment's CLIP embedding projected into 2D UMAP space. Different colors are used to represent distinct concept clusters, with similar visual concepts positioned closer together and dissimilar ones farther apart in this projected embedding space.

Users can interact with the visualization by clicking on any point to reveal an image highlighting the corresponding segment. They can then navigate through additional images in the same concept cluster with the “Previous" and “Next" buttons shown in section D.

3.2 Finding Problematic Data Slices

After generating the concepts, the user can navigate to the Miner interface as shown in Figure 2 to identify problematic data subgroups. As illustrated in section E of Figure 2, users may import the concept-cluster settings generated previously in the Explorer page with the "File" dropdown, pick their class label of interest, specify a maximum slice length, specify the support level for pattern mining, and limit the number of displayed result. Advanced settings

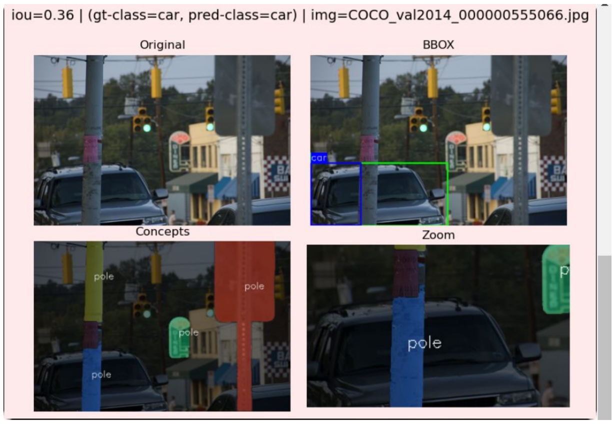


Mining Settings

Label: car
Max Combo: 3
Support: 0.01
Top K: 30
File: 1_demo_base_400.h5.hdf5

Save	items	metrics			
	Itemset	Support	Supp... Count	Accuracy	d_Accu...
	pole=1, road=0	0.0121	34	0.5882	-0.2184
	car_rear=1, road_...	0.0256	72	0.7500	-0.0566
	car=0	0.8793	2469	0.7959	-0.0108

Figure 2: The Miner page displays the mining configurations and mining results.



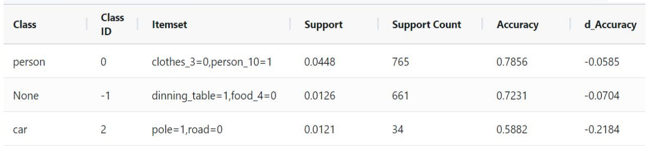
iou=0.36 | (gt-class=car, pred-class=car) | img=COCO_val2014_000000555066.jpg

Figure 3: The Miner visualization panel displays slice images and their corresponding bounding boxes and visual concepts.

allow for result deduplication, concept interaction counting, and inclusion of metadata such as crowding, bounding box area, and aspect ratio.

These various knobs allow practitioners to tailor their mining results to various scenarios. For instance, users can find rare scenarios where the model poorly detects the "car" class by setting "Label" to "car" and support to a small value like "0.005" (0.5%). Similarly, users interested in learning about errors that affect all classes may set "Label" to "None" and toggle for additional metadata with "crowding," "bbox-area," and "bbox-aspect-ratio." Once satisfied with the parameter configuration, users initiate the mining process by clicking the "Submit" button.

The mining results populate the table in section F, where users can click different columns to sort the slicing results based on support, accuracy, or accuracy divergence. Clicking on a specific slice reveals sample images corresponding to that slice, as shown in Figure 3. This visualization offers four perspectives for each



Class	Class ID	Itemset	Support	Support Count	Accuracy	d_Accuracy
person	0	clothes_3=0,person_10=1	0.0448	765	0.7856	-0.0585
None	-1	dinning_table=1,food_4=0	0.0126	661	0.7231	-0.0704
car	2	pole=1,road=0	0.0121	34	0.5882	-0.2184

Figure 4: Itemsets bookmarked by users will be displayed in a table with class and slice data.

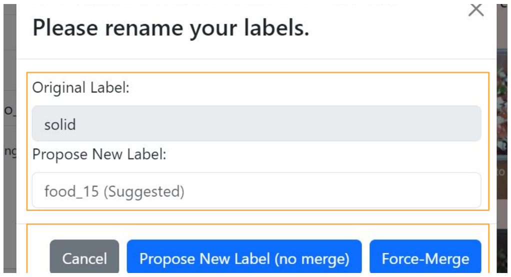
result: the original image, target bounding box pairs, relevant visual concepts and labels, and a zoomed-in view of the concepts.

Users can keep track of their slices of interest with the bookmarking button (Figure 2, section F), so that they may be reviewed later in the Bookmarks Page shown in Figure 4. This feature enables users to reinspect these slices and determine the most effective approach to address them. For instance, some users might fine-tune the model according to the slices. Alternatively, users may choose to employ a different model for the dataset if they find it more robust in handling the identified slice scenarios.

3.3 Concept Cluster Refinement

Slice results may not always be satisfactory for users, and additional refinement may be needed. For example, domain experts may wish to incorporate their own knowledge in the concept definitions. As a result, VCR supports interactive refinement through concept relabeling and concept cluster granularity adjustments.

Concept Relabeling. Visual concept labels generated via CLIP (discussed in §2.1) can be noisy or incorrect. To correct for this and improve user understanding, VCR supports relabeling, where users may change labels to their liking. In addition to renaming cluster in Explorer (Figure 1, section C), users can also perform relabeling in the Miner while exploring data slices. In the Miner, users can double-click on any slice to edit the label text. For example, users may modify the label "pavement_3" to "sidewalk." Then, clicking on the "Save Renaming" button will apply those changes. Additionally, VCR supports label conflict-handling and merging. When practitioners rename a concept cluster to an existing label, VCR notifies users about the conflict, allowing users to either merge the existing visual concepts or propose a new label altogether, as seen in Figure 5.



Please rename your labels.

Original Label:

solid

Propose New Label:

food_15 (Suggested)

Figure 5: Conflicting labels will prompt users to either use a different label or merge labels.

Adjusting Concept Cluster Granularity. Suppose users are unsatisfied with their concept clusters. In that case, VCR allows users to adjust concept granularity with the "Number of Centroids" parameter in Figure 1 section A, which affects all concept clusters simultaneously. For instance, if users find that few or no slice results appear after running data slicing, they may realize that their concept clusters are too fine-grained. In such cases, users can navigate back to the Explorer page, decrease the "Number of Centroids" parameter, and re-run the slicing process. In the future, we also plan to support mechanisms that can adjust individual clusters, such as splitting a cluster into two.

REFERENCES

[1] Rakesh Agrawal, Ramakrishnan Srikant, et al. 1994. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, Vol. 1215. Santiago, Chile, 487–499.

[2] Daniel Bolya, Sean Foley, James Hays, and Judy Hoffman. 2020. Tide: A general toolbox for identifying object detection errors. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 558–573.

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. 2019. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155* (2019).

[4] Vargha Dadvar, Lukasz Golab, and Divesh Srivastava. 2023. POEM: Pattern-Oriented Explanations of Convolutional Neural Networks. *Proceedings of the VLDB Endowment* 16, 11 (2023), 3192–3200.

[5] Terrance De Vries, Ishan Misra, Changhan Wang, and Laurens Van der Maaten. 2019. Does object recognition work for everyone?. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 52–59.

[6] Greg d'Eon, Jason d'Eon, James R Wright, and Kevin Leyton-Brown. 2022. The spotlight: A general method for discovering systematic errors in deep learning models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. 1962–1981.

[7] Xiaoyi Dong, Jianmin Bao, Yinglin Zheng, Ting Zhang, Dongdong Chen, Hao Yang, Ming Zeng, Weiming Zhang, Lu Yuan, Dong Chen, et al. 2023. Maskclip: Masked self-distillation advances contrastive language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10995–11005.

[8] Sabri Eyuboglu, Maya Varma, Khaled Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Ré. 2022. Domino: Discovering systematic errors with cross-modal embeddings. *arXiv preprint arXiv:2203.14960* (2022).

[9] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. 2019. Towards automatic concept-based explanations. *Advances in neural information processing systems* 32 (2019).

[10] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. *arXiv preprint arXiv:2304.02643* (2023).

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 740–755.

[12] Eliana Pastor, Luca De Alfaro, and Elena Baralis. 2021. Looking for trouble: Analyzing classifier behavior via pattern divergence. In *Proceedings of the 2021 International Conference on Management of Data*. 1400–1412.

[13] Gregory Plumb, Nari Johnson, Angel Cabrera, and Ameet Talwalkar. 2023. Towards a More Rigorous Science of Blindspot Discovery in Image Classification Models. *Transactions on Machine Learning Research* (2023).

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[15] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[16] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28 (2015).

[17] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.

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