Information Systems 138 (2026) 102652

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

VCR: Interpretable and interactive debugging of object detection models with visual concepts☆

Jie Jeff Xu [a],*, Saahir Dhanani [a], Jorge Piazentin Ono [b], Wenbin He [b], Liu Ren [b], Kexin Rong [a]

[a] Georgia Institute of Technology, Atlanta, 30324, GA, USA
[b] Bosch Center for Artificial Intelligence (BCAI), Bosch Research North America, Sunnyvale, 94085, CA, USA

ARTICLE INFO

Keywords:
Slice discovery methods
Visual concepts
Human-in-the-loop

ABSTRACT

Computer vision models can make systematic errors, performing well on average but substantially worse on particular subsets (or slices) of data. In this work, we introduce Visual Concept Reviewer (VCR), a human-in-the-loop slice discovery framework that enables practitioners to interactively discover and understand systematic errors in object-detection models via novel use of visual concepts–semantically meaningful and frequently recurring image segments representing objects, parts, or abstract properties.

Leveraging recent advances in vision foundation models, VCR automatically generates segment-level visual concepts that serve as interpretable primitives for diagnosing issues in object-detection models, while also supporting lightweight human supervision when needed. VCR combines visual concepts with metadata in a tabular format and adapts frequent itemset mining techniques to identify common absences and presences of concepts associated with poor model performance at interactive speeds. VCR also keeps humans in the loop for interpretation and refinement at each step of the slice discovery process. We demonstrate VCR's effectiveness and scalability through a new evaluation benchmark with 1713 slice discovery settings across three datasets. A user study with six expert industry machine learning scientists and engineers provides qualitative evidence of VCR's utility in real-world workflows.

1. Introduction

Computer vision models for object-detection and image classification are widely deployed in critical applications such as autonomous driving, medical imaging, surveillance systems, and content moderation. While these models may achieve good average performance, they can exhibit systematic errors on specific subsets (or slices) of data [1–3]. Prior work observed that object-recognition systems perform poorly on household items common in low-income countries, with up to a 20% difference in recognition accuracy between images collected in high- and low-income regions [4]. These discrepancies arise not only from appearance differences within the same object categories but also from contextual variations in which objects typically appear. Similar performance gaps have been observed in various applications including object recognition [5,6], image classification [7,8], and medical diagnosis [9,10]. Detecting these systematic errors could help guide practitioners to update training datasets and mitigate unwanted biases in models [11,12].

For structured, tabular datasets, identifying problematic data slices is relatively straightforward. Tabular slice discovery methods can summarize data slices using attribute–value pairs like $\{age < 20, gender = Female\}$, and mine combinations of attributes that correlate with model errors [2,13–17]. The power of this approach lies in enabling a combinatorial explosion of possible patterns from a limited set of attributes. This raises the question of whether we can use similar image attributes to identify coherent and semantically meaningful data slices in unstructured image datasets. Datasets with rich annotations, like CelebA's 40 labeled attributes per image (*e.g.*, eyeglasses, smiling) [18], illustrate the potential of this approach for vision. Although most real-world datasets lack such structured annotations, images exhibit compositional structure comprising objects, object parts, and spatial relationships, which often correlate with systematic failures. This requires attributes that can capture these compositional elements and are also interpretable to users.

Our key insight is to leverage "visual concepts" as the core, human-interpretable primitives for explaining behaviors of vision models.

☆ This article is part of a Special issue entitled: 'HILDA' published in Information Systems.

* Corresponding author.
E-mail addresses: jxu680@gatech.edu (J.J. Xu), saahir@gatech.edu (S. Dhanani), jorge.piazentinono@us.bosch.com (J.P. Ono), wenbin.he2@us.bosch.com (W. He), liu.ren@us.bosch.com (L. Ren), krong@gatech.edu (K. Rong).

https://doi.org/10.1016/j.is.2025.102652
Received 16 June 2025; Received in revised form 11 November 2025; Accepted 2 December 2025
Available online 12 December 2025

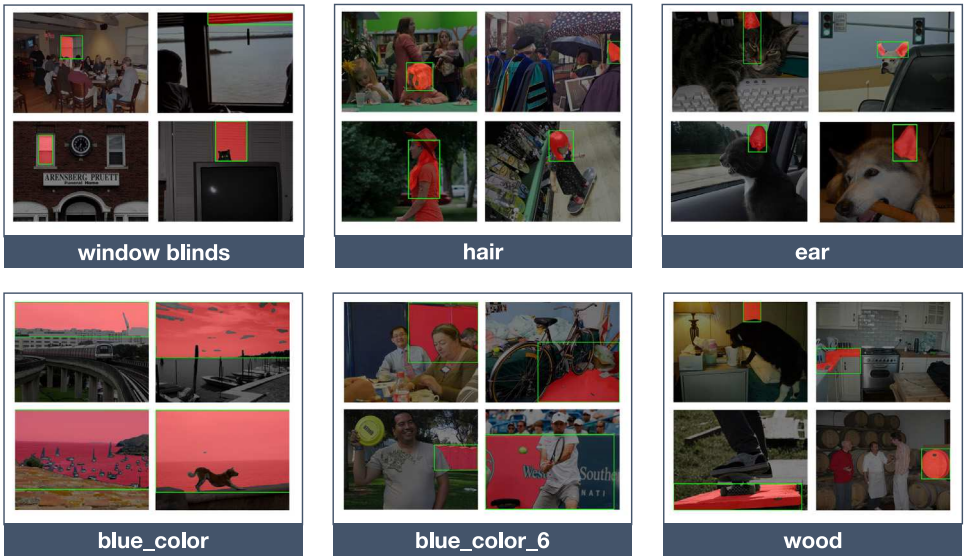


Fig. 1. Example visual concepts that represent objects, parts of objects, as well as abstract properties such as color and material. Visual concepts in each image are highlighted in red and their corresponding labels are attached below. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

While visual concepts have various definitions across the literature [19–24], in this work we define them as *semantically meaningful and frequently recurring image segments* that can represent objects, object parts, or abstract properties like color and material, as shown in Fig. 1. These concepts serve as suitable primitives for slice discovery for two reasons: (1) their semantic meaning allows practitioners to define and understand slices in intuitive terms, similar to tabular attributes, (2) their recurrence across the dataset enables the discovery of systematic patterns rather than isolated failures.

We introduce Visual Concept Reviewer (VCR), framework that operationalizes this idea for discovering systematic errors in object-detection models. VCRbuilds upon recent advances in vision foundation models [25,26] to automatically extract and label visual concepts, such as "pole", "wheel", and "pedestrian" in a driving dataset, while still allowing light human supervision for refinement. It then presents these concepts to a user through an interactive interface, which enables users to explore extracted concepts through 2D visualizations and adjust concept granularity and definition, and guide the analysis process using their domain expertise. VCR augments rather than replaces human experts, automating the identification of salient visual patterns in large datasets while keeping humans in the loop for interpretation and refinement.

To discover problematic data slices, VCR analyzes the interactions between these visual concepts and object-detection models' bounding box predictions. For example, VCR may explain a set of poor detection results for the "car" class with the presence of the visual concept "pole" that occludes the view. While VCR builds on frequent itemset mining techniques to find combinations of visual concepts highly correlated with poor model performance, a key difference is its ability to consider the *absence* of concepts, which can reveal critical failure modes. For example, if a model relies on the concept "wheel" to identify cars, its absence could lead to misclassifications. However, naïvely supporting concepts absences could lead to a combinatorial explosion of trivial mining results dominated by absences, as concept absence is common. To improve the interactivity of the workflow, VCR introduces pruning optimizations based on mutual information analysis, which speeds up the mining performance by up to two orders of magnitude in our experiments.

In summary, we contribute:

- VCR, a human-in-the-loop slice discovery framework for object-detection models that combines automated visual concept extraction with interactive exploration to uncover systematic model failures.
- A new evaluation benchmark for slice discovery methods in object-detection, which includes 1713 slice discovery settings across three widely used datasets. On this benchmark, VCR consistently outperforms existing methods in recovering ground truth error patterns.
- A user study with six industry machine learning scientists and engineers, providing qualitative evidence that VCR supports real-world model understanding workflows and helps users identify meaningful error patterns.

2. Visual concepts

In this section, we introduce the design considerations behind the notion of visual concepts used in this work (Section 2.1) and describe a preprocessing pipeline that automatically extracts visual concepts, while allowing light human supervision for refinement (Section 2.2).

2.1. Definition and design considerations

Visual concepts have been defined in various ways across the literature. Some methods require users to provide labeled examples to define concepts of interest [23,27,28] or interactively collaborate with the system to identify concepts [29–32]. Others consider neurons in deep neural networks with similar activation maps as defining visual concepts [24,33,34] or treat semantically meaningful regions of images as visual concepts [35].

In this work, we define visual concepts as *(1) semantically meaningful* and *(2) commonly occurring* segments of natural images. These two properties of visual concepts make them suitable primitives for identifying and explaining systematic errors: semantically meaningfulness improves the interpretability of results, while common occurrence enables the capture of systematic behaviors.

Semantically Meaningful. The first aspect of our definition, semantic meaningfulness, is crucial for visual concepts to serve as effective building blocks for image understanding. This requires image decomposition methods that respect object boundaries. Grid-based decomposition

methods [36] offer an overly simplistic approach by breaking images into individual squares, failing this requirement since these boundaries rarely align with object boundaries; objects may span multiple cells and be fragmented into semantically ambiguous portions. We instead leverage image segmentation techniques, which are explicitly designed to group pixels into coherent, semantically meaningful regions. Using segmentation masks as the foundation for visual concepts aims to align each concept with a meaningful object or visual pattern, though segmentation quality may vary depending on image characteristics and model performance.

Common Occurrence. The second aspect, common occurrence, is important as it allows us to generate a shared "visual vocabulary" to describe image datasets. Similar segments often reappear in different contexts, varying in location, size, and specific content. To capture the natural recurrence of similar segments across images, we use clustering to group semantically similar image segments into visual concepts. Clustering, therefore, allows concepts to naturally emerge from the dataset rather than being limited to predefined categories.

Fig. 1 shows examples of visual concepts extracted from the MS COCO dataset [37]. Segments highlighted in red are sampled from the corresponding visual concept. Empirically, we have found that concepts could represent whole objects (e.g., "window blinds", "pizza", "bicycle"), object parts (e.g.,"arm", "hair", "animal ears"), or properties like colors (e.g.,"blue_color", "dark_color") and material (e.g., "wood", "concrete"). Due to the nature of clustering, the boundaries between concepts are sometimes blurred. For instance, the "blue_color" concept shown in Fig. 1 contains many segments of the sky, and is close to concepts labeled as "sea", "sky" and "cloud" in the embedding space. A coarser clustering granularity may merge these separate concepts into a single cluster.

2.2. Visual concept generation pipeline

With these design principles for visual concepts in mind, we present a preprocessing pipeline to generate visual concepts from image datasets using pre-trained vision foundation models. The pipeline consists of four steps: segmentation, embedding, clustering, and an optional labeling step.

(1) **Segmentation.** The first step is to extract meaningful segments from each input image using an image segmentation model. We specifically choose Meta's Segment Anything Model (SAM) [25] for this task, as it has an implicit notion of "objects" derived from its training process. Given a prompt (e.g., one or more points), SAM returns a segmentation mask containing at least one of the objects referred to in the prompt. For example, given a point on a backpack, SAM might return a mask for either the backpack or the person wearing it—both semantically meaningful entities. After segmentation, we filter out segments that occupy less than one percent of the total image area, as they are often noisy and lack meaningful content.

(2) **Embedding.** The next step is to capture the semantic meanings of these image segments. We experiment with two pre-trained vision foundation models that generate embeddings at different resolutions: CLIP [38] produces image-level embeddings and MaskCLIP [26] yields pixel-wise embeddings.

For CLIP, we crop each segmentation mask to its minimum bounding box, gray out non-segment pixels, and embed the result. However, we observed that this approach performs poorly, aligning with recent findings that pre-trained CLIP performance degrades on masked images due to distribution shifts [39]. We therefore focus on MaskCLIP which generates pixel level embeddings. We align these embeddings with SAM segments by resizing the segments to match each model's output dimensions.

(3) **Clustering.** We perform K-means clustering on the segment-level embeddings to derive visual concepts, where segments grouped into the same cluster represent the same visual concept. Importantly, the number of clusters is a user-controlled parameter that allows practitioners to adjust the granularity of visual concepts based on their debugging needs and domain expertise. For instance, practitioners debugging autonomous vehicle models might prefer finer-grained concepts to distinguish between different types of road signs, while those working on general object-detection might use coarser granularity for broader categorical understanding. Empirically, we found that hundreds to thousands of concept clusters generally provide a good balance between comprehensive coverage and manageable exploration for practitioners. VCR's interactive interface allows users to dynamically adjust this granularity and merge or split concepts based on their domain knowledge, ensuring the visual vocabulary aligns with their specific debugging priorities. This clustering step creates a shared "visual vocabulary" that captures recurring patterns across the dataset while remaining interpretable and actionable for human analysts.

(4) **(Optional) Labeling.** To further improve interpretability, we can leverage the multimodal nature of CLIP-based segment-level embeddings to automatically assign text labels to concept clusters. This is achieved by pairing each cluster center with the text label having the closest embedding distance. The input labels can be sourced independently, such as from a list of commonly used English nouns or domain-specific labels provided by users or large language models.

When multiple concepts map to the same label, we resolve conflicts by appending numbers to create subcategories (e.g., "tree_1", "tree_2", and "tree_3"). This reflects either meaningful distinctions between concepts, or overly fine clustering granularity. For instance, in Fig. 1, the concepts "blue_color_6" and "blue_color" represent distinct visual patterns: the former includes background objects like ad banners and portable toilets, while the latter contains segments of sky and sea.

The quality of generated visual concepts depends on the performance of the underlying segmentation and embedding models. To allow VCR to benefit from ongoing advances in the vision community, its concept generation pipeline is intentionally modular–practitioners can swap in newer architectures without requiring fundamental changes to the framework. Additionally, VCR's interactive workflow (detailed in Section 3) allows practitioners to refine the visual vocabulary based on their domain expertise, further compensating for limitations in automated segmentation or embedding.

3. VCR: Overview and user workflow

Building on the structured representation of images provided by visual concepts, we develop VCR, an interactive slice discovery framework that identifies and explains systematic errors in object-detection models. VCR focuses on the object-detection task, as it is the most widely used computer vision task in practice, although visual concepts could also be used for other tasks such as image classification. In this section, we present background on the object-detection task, and introduce VCR's user interface and workflow.

3.1. Background: Object-detection

Object-detection models analyze images to identify and locate objects, providing predictions in the form of bounding boxes and associated class labels. The accuracy of these models is evaluated by comparing their predictions to ground-truth data, focusing on two key aspects: class prediction accuracy and bounding box alignment. The quality of bounding boxes is often assessed using the Intersection over Union (IoU) metric, which measures the overlap between two bounding boxes by dividing the area of intersection by the area of their union. A key step in this evaluation process is bounding box matching, which associates each predicted box with a corresponding ground-truth box. Different systems implement this matching task differently: some prioritize maximizing IoU (*e.g.*, using the Hungarian algorithm), while others give preference to class label confidence.

Object-detection models' dual objectives of ensuring both correct classification and accurate object localization lead to several potential error types:

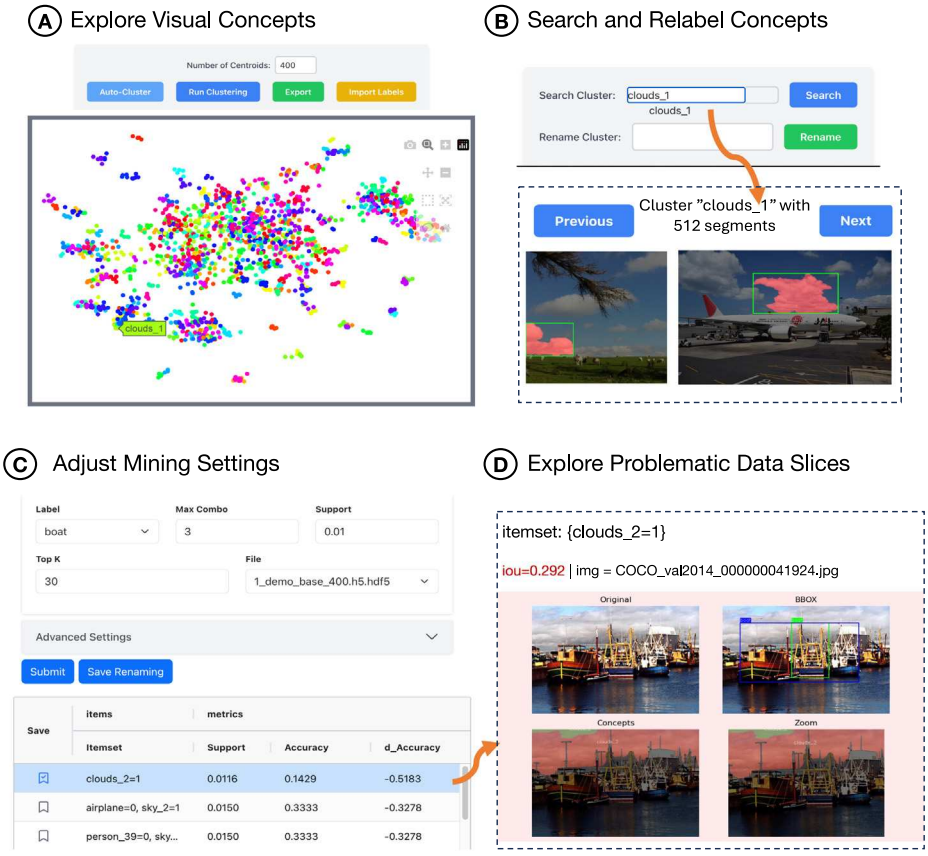


Fig. 2. VCR's user interface allows users to adjust mining settings, explore data slices and visual concepts, as well as perform refinement such as relabeling a concept and merging clusters. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- *Classification error:* The bounding box is localized correctly (IoU greater than some threshold) but classified incorrectly.
- *Localization error:* The bounding box is classified correctly, but the IoU with ground truth is low, indicating poor localization.
- *Background error (false positive):* The model incorrectly detects the background as objects.
- *Missed ground truth (false negative):* Ground truth objects are undetected, not covered by classification or localization errors.

These error types are essential considerations in evaluating and improving object-detection models [40]. VCR supports all of the above error types.

3.2. User interface and workflow

Fig. 2 shows VCR's user interface and workflow, designed to support practitioners in iteratively building understanding of their model's behavior through visual concept exploration and refinement.

The Concept Exploration view (part A) displays the automatically extracted visual concepts in a 2D UMAP projection [41], helping users understand the overall concept space. Each color represents a different concept, and hovering over a dot reveals the automatically generated concept label. Users can experiment with concept granularity by changing the total number of clusters using the top panel.

Users can also delve into specific concept clusters by clicking on them, which displays sample image segments belonging to that concept. Part B shows example segments from a concept labeled as "clouds_1". Upon inspection, the user may realize that this cluster contains similar segments compared to the "clouds" cluster, so she can merge these two concepts by renaming them the same label. Our demonstration paper [42] provides additional details on handling label conflicts during merging.

After exploring and refining concepts, users export their customized concept set to the Mining Interface (Part C), where they can adjust settings such as filtering bounding boxes via class labels, support threshold (minimum slice size) and error type according to their specific debugging goals. The mining results appear in a sortable table, where each row represents a data slice summarized by how bounding boxes interact with surrounding visual concepts and their metadata. For example, the highlighted row $clouds\_2 = 1$ represents bounding boxes labeled as boats that interact with one segment of a clouds concept. This slice shows an accuracy difference of −0.52, meaning that the average IoU for this data slice predictions is 0.52 lower than the average of the entire dataset. Users can sort the table to prioritize slices by different criteria (e.g., size of slice, average accuracy), and save interesting slices for later analysis in the Bookmark Page.

Finally, clicking on a specific slice reveals sample images (part D) with four complementary views: the original image, bounding box pairs, relevant visual concepts, and zoomed concept details. This multi-view visualization helps practitioners understand why the model fails on particular slices and assess whether the discovered patterns represent genuine systematic errors or spurious correlations. Beyond these views, an additional interface (Appendix D) for exploring concept absences helps users interpret these often challenging cases. This interface allows users to compare an itemset containing a concept absence with its counterpart in which the same concept is present. Presenting these cases side by side helps contextualize the absence and clarify its impact on model behavior.

4. Concept-based slice discovery

Internally, VCR leverages frequent itemset mining techniques to identify correlation between problematic model behaviors and the

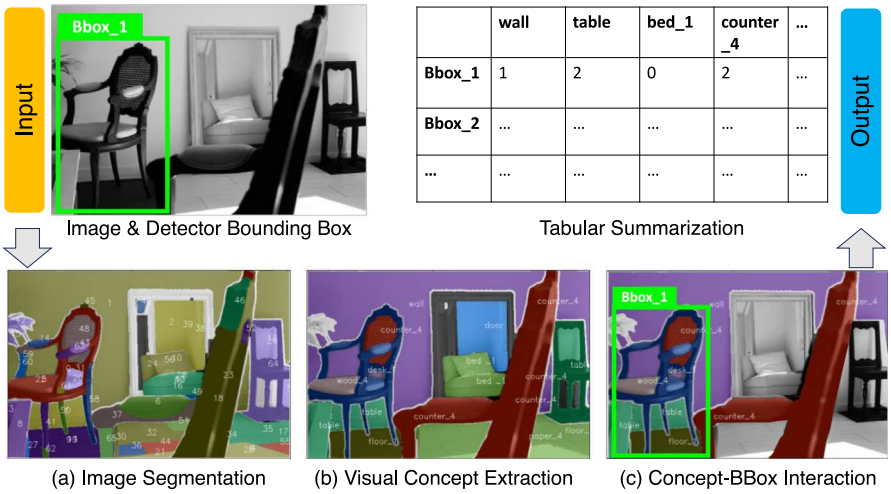


Fig. 3. VCR summarizes the interaction between visual concepts and the object-detection model's predictions in a tabular format. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

presence and absence of visual concepts. Section 4.1 describes VCR's automated slice discovery pipeline. While classic frequent itemset mining that focuses only on frequently occurring items, accounting for the *absence of items* is equally valuable in our application. However, naïvely supporting concepts absences leads to a combinatorial explosion of mining results dominated by absences, as concept absence is common. VCR introduces new pruning optimizations based on mutual information analysis to address the scalability challenges of supporting concept absences (Section 4.2).

4.1. VCR: *Slice discovery pipeline*

VCR operates in three main stages: concept extraction, tabular summarization, and concept mining. In the preprocessing step, VCR extracts visual concepts for the input dataset using methods described in Section 2.2. Fig. 3(a) shows an example segmentation obtained using SAM, and Fig. 3(b) shows that the input image can be represented by a set of visual concepts such as "wall", "bed", "table", and "wood". These visual concepts serve as primitives for analyzing an object-detection model's performance.

Tabular Summarization. VCR uses visual concepts to contextualize the object-detection model's predictions. For each pair of predicted and ground truth bounding boxes, VCR identifies all visual concepts that have significant pixel overlap with the boxes. A fixed padding (*e.g.*, 50 px) is applied around each box to capture additional nearby concepts, as bounding boxes are generally tightly fitted around the object. For instance, in Fig. 3(c), two segments (in teal color) from the "table" concept overlap with the bounding box. These interactions are summarized in a tabular format, where each row corresponds to a pair of matched bounding boxes, each column represents a different visual concept, and cell values indicate the number of image segments from each concept interacting with each bounding box pair.

Concept Mining. Given the tabular summarization, VCR uses the Apriori algorithm to identify visual concept patterns that correlate with poor model performance. We chose Apriori because its level-wise bottom-up approach allows us to easily enforce constraints on concept absences paired with presences at each support threshold, bubbling up incrementally for increasing itemset lengths. This property is particularly valuable for our pruning optimizations (Section 4.2), where we disallow multiple absences in size-2 itemsets to prevent their propagation to longer itemsets. Alternative frequent pattern mining algorithms like FP-growth [43], while often faster for traditional mining tasks, build compressed tree structures (FP-trees) that make it challenging to selectively control the generation of specific itemset combinations based on absence constraints during tree construction. VCR considers various error types, including classification error, localization error, background error (false positive), and missed ground truth (false negative), as discussed in Section 3.1. Given an error type and an IoU threshold (default at 0.5 as in [40,44]), VCR marks a subset of the bounding box pairs as problematic according to the error metrics. VCR then outputs common patterns among problematic predictions as frequent itemsets, ranked by accuracy divergence—a metric that quantifies how much a slice's performance deviates from the overall average [14]. For a slice S with error metric M, accuracy divergence is calculated as $\Delta_{acc}(S) = M(S) - M(avg)$, where negative values indicate worse-than-average performance. This ranking helps practitioners prioritize the most problematic systematic errors for investigation. Users can specify a minimum slice size (minimum support) parameter, and VCR identifies all data slices above this threshold.

VCR supports itemsets that are conjunctions of predicates with both image metadata attributes and attributes enabled by visual concepts. These include bounding box statistics (relative size, aspect ratio, and position), crowding information (number of nearby overlapping bounding boxes), image metadata (class labels, time of day, location), and the count of each type of visual concept that interacts with the model prediction. Numeric attributes, such as bounding box area, are automatically discretized into ten bins using quartiles. Then, an example slice for a barely visible car in a busy intersection may look like $\{gt\_bbox\_area \in [0, 0.25], crowding \in (5, 10]\}$ where the bounding box area of the car is in the bottom twenty-five percentile and there are five to ten other cars surrounding it.

Limitations. Co-occurrence-based concept mining can surface spurious correlations where identified patterns reflect coincidental associations rather than true causal relationships. Visual concepts should therefore be treated as discovery aids that guide practitioners toward error patterns and require expert validation. As shown in our user study (Section 5.3), spurious correlations that group together similar error patterns can still serve as effective starting points for inspection.

4.2. *Challenge: Supporting concept absences*

In the context of concept mining, VCR differs from classic frequent itemset mining techniques by considering not only the presence of items but also their *absence*. This is crucial for explaining the behavior of object-detection models, as the absence of certain concepts can be just as informative as their presence. For instance, a model trained to

identify cars based on the presence of wheels may fail when wheels are not visible in an image.

However, supporting concept absence mining introduces two significant challenges: scalability and result redundancy. The scalability issue arises from the sparsity of concepts. For a dataset with many concepts, most concepts would be absent for a given predicted and ground truth bounding box pair. Therefore, naïvely treating absent concepts as frequent items would lead to a combinatorial explosion of results dominated by concept absences. For example, POEM experienced significant performance degradation in mining beyond 15 concepts when considering absence [24]. Second, concept absence introduces additional redundancies in the mined itemsets. For example, $\{sky = 1, sea = 0\}$ might represent a similar set of bounding-boxes as $\{sky = 1, land = 0\}$. Presenting multiple itemsets that essentially represent the same data can be confusing for users. Furthermore, accumulating concept absences in itemsets can produce uninformative results like $\{sky = 1, sea = 0, land = 0, toaster = 0, potato = 0\}$, which add little value to the analysis.

Mutual Information Analysis. To understand when concept absences are meaningful, we analyze the information gain between two itemsets using normalized mutual information (NMI).

Consider two itemsets A and B where B is a subset of A (*e.g.*, B contains one additional item on top of A). We want to evaluate the additional information B provides given A using NMI, defined as $NMI(A, B) = \frac{2I(A;B)}{H(A)+H(B)}$, where $H(A)$ and $H(B)$ are the entropies of A and B, and $I(A; B)$ is their mutual information. NMI is scaled between 0 and 1, with 1 indicating complete dependence between two events. A higher NMI suggests that B does not add much new information beyond A.

Proposition 4.1. *Consider two itemsets A and B. Assume that B is a subset of A, and that $P(A) = p, P(B) = q$. $NMI(A, B)$ decreases as $p - q$ increases.*

The proof is available in Appendix A. We note two special cases of the proposition. When A and B are identical ($\delta = 0$), $NMI(A, B) = 1$. When B is rare ($q \to 0$), $NMI(A, B) \to 0$, regardless of A's probability.

This analysis helps distinguish between meaningful and trivial concept absences. Specifically,

- Frequent co-occurrence: Suppose $A = \{sea = 1\}$ and $B = \{sea = 1, boat = 0\}$. Since "sea" and "boat" frequently co-occur (large δ), B might be worth investigating.
- Rare co-occurrence: Suppose $A = \{sea = 1\}$ and $B = \{sea = 1, carrot = 0\}$. Since "carrots" rarely appear with "sea" (small δ), B is not likely to add much new information on top of A. In fact, if "carrots" and "sea" never co-occur, A and B are identical and redundant.
- Rare concept: Suppose $A = \{sky = 1\}$ and $B = \{sky = 1, unicorn = 1\}$. Rare concepts could be informative if it occurs. However, their absences are not interesting: the absence of a unicorn in the sky is so common that it does not provide meaningful information beyond knowing there is sky in the image. We do not consider very rare concepts, as each itemset needs to pass the minimal support threshold.

Pruning and Duplication Optimization. Based on the analysis above, we introduce two optimizations that significantly improve the mining performance for concept absences, as demonstrated in Section 5.2.

In our first optimization, we implement a threshold-based filtering mechanism, where an absence is only included if it contributes a threshold amount of new information. In the implementation, we use the relative change in support as an approximation to NMI, as it is computationally more efficient and strongly correlated with NMI. Given itemset A and concept absence $\{c\}$, the relative support change in $B = A \cap \{c\}$ is $\Delta_{\text{Sup}}(A, B = A \cap \{c\}) = Sup(A \cap \{c\})/Sup(A)$.

Empirical Calibration. We validate the above approximation by generating 100 synthetic parent–child itemset scenarios at varying densities p (the percentage of data covered by the parent itemset), with each scenario evaluated on 100,000 samples. For each scenario, we compute the exact NMI and relative support change. Fig. 4 shows the resulting curves stratified by p. The inverse relationship is nearly linear across all densities, with coefficient of determination R^2 ranging from 0.701 for dense itemsets ($p = 0.9$) to 0.997 for very sparse ones ($p = 0.1$). Since sparse itemsets ($p \leq 0.5$) dominate our mining results and achieve $R^2 \geq 0.955$, the approximation is highly reliable in practice.

For our second optimization, we restrict itemsets from containing multiple concept absences. We expect important multiple absences to show up as single absences in other itemsets, so this constraint maintains similar itemset quality while significantly reducing computational overhead, particularly addressing the problem of accumulating absences. We implement this restriction when generating itemsets of size two, which effectively precludes the creation of itemsets with multiple absences in itemsets of arbitrary length, per the Apriori principle.

Finally, we post-process the itemsets to remove near-duplicate results.

We use a lightweight, greedy deduplication algorithm that aims to preserve interesting itemsets while maximizing the diversity. The algorithm iterates through the itemsets in order of decreasing accuracy divergence, marking the bounding box pairs covered by each included itemset. We only include new itemsets when they contain a significant portion of unique bounding box pairs, controlled by the parameter $\delta \in [0, 1]$. In practice, we set $\delta = 0.5$ by default. Example results illustrating the greedy deduplication step and its diversity impact are provided in Appendix B.

5. Evaluation

We create a large-scale evaluation benchmark comprising 1713 slice discovery settings across three widely used datasets, enabling quantitative comparisons of slice discovery methods in object-detection tasks. VCR consistently outperforms baselines in identifying problematic subgroups and that the pruning optimization significantly speeds up discovery (Section 5.2). To complement the quantitative evaluation, we conduct a user study with six industry machine learning scientists and engineers to provide qualitative insights into VCR's real-world usability (Section 5.3).

5.1. Evaluation methodology

Few real-world object-detection datasets specify data slices where a model systematically underperforms. Following existing practices [1, 45,46], we programmatically generate 1713 slice discovery settings to enable quantitative performance comparison. Table 1 summarizes our evaluation benchmark.

Table 1
Overview of our slice discovery evaluation benchmark.

Datasets	Color	Aspect ratio	Size	Semantic	#Settings
COCO	287	156	153	184	780
Visual Genome	217	143	140	113	613
BDD 100K	100	65	75	80	320

Slice discovery settings. We evaluate on three widely-used datasets in object-detection tasks: COCO 2014 [37], Visual Genome [47], and BDD 100K [48]. We consider two types of reasons that cause the model to underperform: metadata-based and content-based. Slices derived from metadata can exhibit better visual consistency, while those derived from content will have better semantic coherency. Accordingly, we create four error scenarios based on:

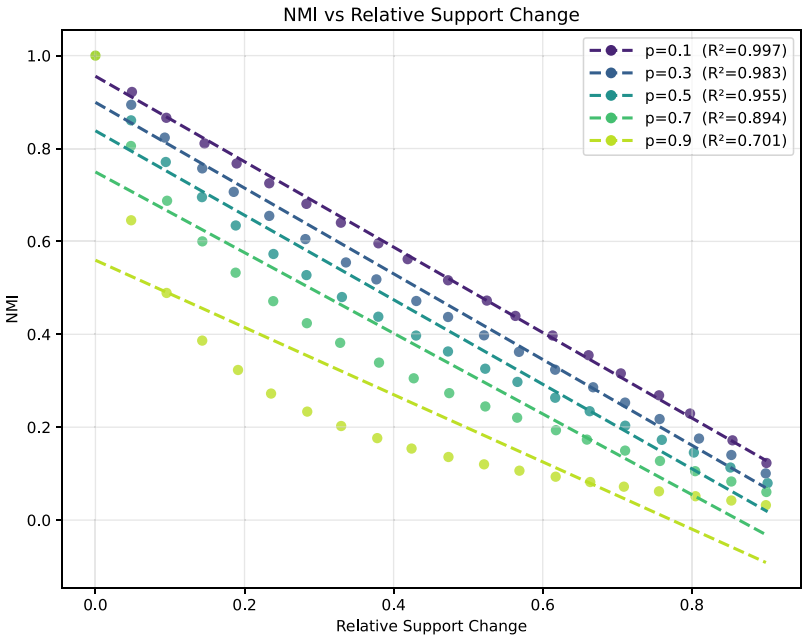


Fig. 4. Relationship between exact NMI and relative support change across multiple itemset densities p. Least-squares fits (dashed) highlight the strong monotonic trend that justifies using support change as a computationally efficient proxy, especially for sparse itemsets.

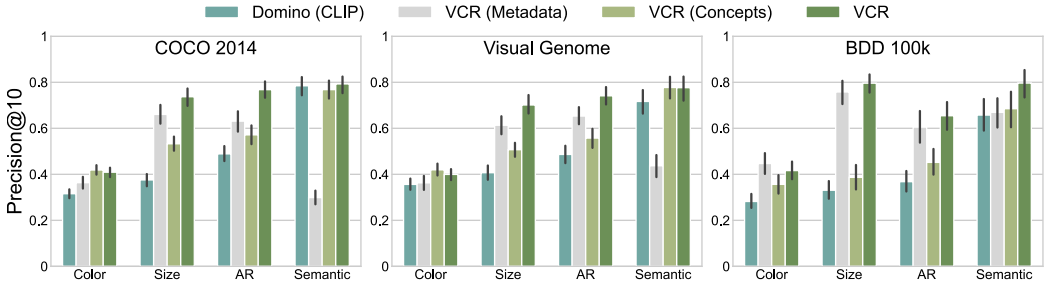


Fig. 5. VCR leads to consistent improvement in slice discovery performance compared to baselines across three datasets and four error scenarios.

- *Color*: Ground truth bounding boxes are resized into 50 × 50 squares, followed by K-Means clustering on raw pixel values to assign them into different color clusters. We use 20 color clusters for each class of objects, and each cluster forms a slice.
- *Size and Aspect Ratio*: Bounding boxes are sorted into bins based on width*height for size and width/height for aspect ratio. We use a randomly chosen number of bins between 5 and 15 for each object class, and each bin forms a slice.
- *Semantic*: Inspired by [46], we extract semantically coherent slices such as "[object class] next to [setting/object]" (*e.g.*, "person next to water"). For COCO and Visual Genome, we use their image captions, embed them with CLIP embeddings, and retrieve up to 500 images closest in the embedding space to our target prompts. For BDD, we use the provided annotations to define slices based on weather conditions, time of day, and scene (*e.g.*, highway, residential); VCR does not directly use these metadata as slicing dimensions.

We discard slices with fewer than 25 samples. Once a slice is generated, we synthetically increase its localization error rate by perturbing the predicted bounding box location to lower the IoU. We model the IoU errors as a Gaussian distribution centered at 0.4, just below the standard 0.5 IoU error threshold. This allows us to create ground truth problematic slices for evaluation. While we focus on localization error, other error types can be supported similarly.

Our default object detector is Faster-RCNN from MMDetection [49, 50] trained on the MS-COCO Dataset's 2017 train split. We provide additional details of the slice discovery setting generation, as well as samples of ground truth slices (Appendix C).

Methods of Comparison. We consider the following methods:

- Domino [1]: Similar to VCR, Domino leverages external, pre-trained models to generate image embeddings for slice discovery. Specifically, we configure it to use CLIP embeddings and set up its Gaussian Mixture Model with 100 clusters and $\gamma = 40$ after hyperparameter tuning. We also provide Domino IoU values as its error metric.
- VCR (concept): VCR using concept interactions as the only slicing dimension (i.e. without metadata). This baseline is similar to POEM [24], which also only uses visual concepts as explanation primitives.
- VCR (metadata): VCR with only metadata attributes in the itemsets. This baseline is similar to SliceTeller [51], which slices datasets based on predefined metadata attributes.
- VCR: By default, we use 500 concepts, a support count threshold of 10, limit the maximum itemset length to 3, and the greedy de-duplication algorithm with an overlap threshold $\delta = 50\%$. For models, VCR uses SAM's ViT-L for segments and MaskCLIP Vit-B/16 for pixel-level embeddings. We show that VCR's performance is not sensitive to specific parameters in Appendix B.1 of the technical report [52].

Since previous works do not support object-detection tasks, we simplify the tasks by cropping images from ground-truth bounding-box annotations and treating them as image classification tasks for baselines. We apply additional pixel padding to enhance the quality of image crops, which provides important contextual information and improves the performance of baselines.

Evaluation Metric. We use Precision@k to evaluate the performance of slice discovery methods. This metric measures the proportion of the top-k most erroneous samples in a discovered slice that are also present in the ground truth problematic slice. We set $k = 10$ and report the maximum precision across the top 10 discovered slices for each method. For VCR, samples are ranked by their error contribution. For Domino, which uses a Gaussian Mixture Model, samples are ranked by their membership probability to a given slice.

5.2. Quantitative comparison

Slice Discovery Performance. Fig. 5 summarizes the performance of VCR and baselines across all slice discovery settings. Overall, VCR consistently outperforms Domino, and both visual concepts and metadata contribute meaningfully to VCR's performance. For the metadata tests, VCR's mean difference with Domino in precision score is 0.244 (60.9% increase), 0.206 (48.2% increase), and 0.292 (87.8% increase) in the COCO, Visual Genome (VG), and BDD datasets, respectively. For the semantic tests, VCR's mean difference with Domino in the precision score is .008, 0.100 (14.0% increase), and 0.170 (25.9% increase) in the three datasets, respectively.

Domino sees its best performance on the semantic slice tests and is not far behind VCR. This is expected as Domino uses CLIP embeddings of images to form semantically coherent clusters. Understandably, VCR (metadata) cannot differentiate semantic attributes and experiences decreased precision in most semantic settings.

However, in BDD's semantic test cases, we find VCR (metadata) can perform relatively well. While semantic tests in COCO and VG were derived from classes' relations to other objects or settings (*e.g.*, indoors), BDD's semantic tests featured both concrete dimensions with "scene" as well two abstract ones, "timeofday" and "weather". We find that VCR (metadata) achieves higher precision than Domino in "time of day" and "weather", only losing in the scenes test. When combined with VCR (concepts), metadata enables a significant increase in precision. This highlights the importance of metadata attributes even in the semantic setting.

For metadata slices on bounding box size and aspect ratio, VCR sees the biggest improvements over Domino, outperforming by 0.255–0.465 precision points in the three datasets. This is because VCR utilizes bounding box statistics as slicing dimensions, while Domino's semantic-based slicing is a poor fit for these object-detection-specific error scenarios. For color clusters, all methods exhibit relatively poor performance since neither metadata nor visual concepts explicitly capture the concept of color. Color clusters also tend to be more noisy compared to other test cases, particularly as we directly utilized raw pixel values. However, for specific classes like traffic lights and umbrellas, coherent color clusters (e.g., Fig. C.11) could be formed. VCR still outperforms Domino by 0.04 to 0.13 precision points across datasets.

Concept Mining Scalability. We compare VCR against two representative tabular slice discovery frameworks, DivExplorer [14] and SliceLine [15], using their open-source Python implementations [53, 54]. DivExplorer supports Apriori and FP-growth, while SliceLine uses a linear-algebra-based method. We use binary concept presence/absence data extracted from the COCO dataset with 100,000 rows and 200 columns (concepts). All methods use a support threshold of 0.03 and generate itemsets with presence/absence up to length 3. Fig. 6 reports the mining runtime versus concept count in log scale, averaged over five runs. DivExplorer with FP-Growth takes over an hour at 75 concepts, and SliceLine triggers the out-of-memory killer at 100 concepts.

Table 2
Effect of number of clusters (k) and deduplication threshold (δ) on VCR's precision on the COCO dataset. The rows in bold represent the default experiment configuration.

Configurations	Color	Aspect ratio	Size	Semantic
k = 500	0.409	0.737	0.767	0.792
k = 400	0.408	0.728	0.746	0.762
k = 300	0.409	0.722	0.732	0.749
k = 200	0.405	0.750	0.764	0.751
δ = 50%	0.409	0.737	0.767	0.792
δ = 25%	0.411	0.740	0.768	0.790
δ = 0% (No Dedup)	0.324	0.664	0.674	0.687

In contrast, VCR finishes within 10 s even at 200 concepts, indicating at least two orders of magnitude speedup, mainly due to the absence pruning optimization.

Sensitivity Analysis. Table 2 presents VCR's performance across varying deduplication thresholds (δ) and concept cluster counts (k). While deduplication enhances VCR's performance, VCR is not sensitive to the overlap threshold δ. VCR is also robust to changes in the number of clusters. Moreover, users can dynamically adjust concept counts and labels through our concept explorer interface.

5.3. Qualitative evaluation with domain experts

To gain qualitative insights into how practitioners leverage VCR's interactive capabilities to discover model failures, we conducted user studies with six expert users who used the system to analyze systematic errors in an object-detection model trained on the COCO dataset.

Expert User Demographics. We interviewed six expert users (industry ML scientists and engineers). All experts have two or more years of experience with machine learning (5.67 ± 2.49 years) and are familiar with object-detection (four have trained detection models before, and two have used but not trained these models). Five of the experts have a PhD in STEM, and one has a Masters Degree. The experts are not authors of this paper.

Interview Protocol. Each interview lasted 45 min and proceeded as follows: First, the user filled out a demographics questionnaire (5 min). Next, we demoed the system capabilities using the object "car", allowing the experts to identify the model problems in the data slices and answer any questions they had (10 min). We then asked the experts to use VCR to identify and understand the model limitations, *i.e.*, problematic data slices, in other objects (20 min). Finally, we asked them for feedback, including positive aspects, negative aspects, and points of improvement (10 min). We also used a 5-point Likert scale questionnaire to assess the perception of the system's functionality and usability.

Expert user analysis and insights. While our sample size limits generalizability, the expert sessions provided valuable qualitative insights into VCR's usage patterns and utility in practice. We highlight the key themes that emerged.

The experts evaluated two to three objects per session (see Fig. 7). A shared strategy involved initially examining the worst-performing slices. This allowed them to pinpoint where the models struggled with object-detection. Next, they inspected the slices where the models outperformed the average, indicating scenarios where the model could easily detect the objects. Here, we list some of their findings:

1. *Occlusion* was frequently listed as a fundamental reason for subpar detection performance. This issue was identified in the objects "car" and "chair". During the system demo using the "car" object, experts immediately found that the poorest performing data slice was defined by the itemset $\{pole = 1, road = 0\}$ ($\Delta_{acc} = -0.218$), indicating the presence of a pole near the undetected object in the image. Upon

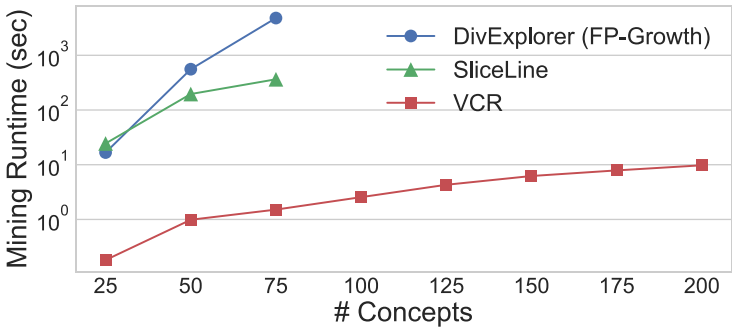


Fig. 6. VCR's absence pruning optimization enables significant speedups compared to alternative mining approaches.

Fig. 7. Example data slices investigated by expert users. Concepts are highlighted in bright colors. The green bounding box indicates ground truth, and the blue bounding box indicates prediction. Top: data slice identified for the object "car", where poles obstruct the detection of the object. Bottom: data slice identified for the object "book", where crowding hampers the detection of the object. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

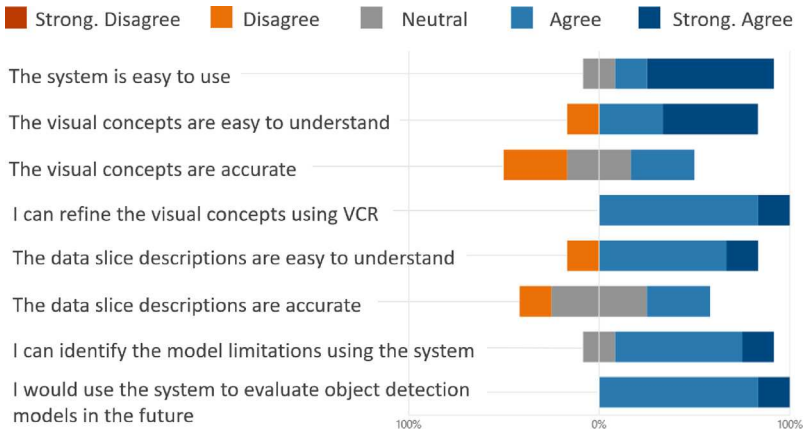


Fig. 8. Expert ratings of VCRacross usability dimensions ($n = 6$), using a 5-point Likert Scale questionnaire.

further examination, it was observed that the pole was obstructing the car in the picture. Similarly, when the experts were inspecting the "chair" object, they found that undetected chairs were frequently due to occlusion by babies or toddlers, as indicated in the slice $\{baby = 1, diningtable = 0\}$ ($\Delta_{acc} = -0.248$).

2. *Crowding* was also a frequent error identified by the experts. For example, when exploring the data slices belonging to "book" object, they noticed that the slice $\{bookshelf = 1\}$ ($\Delta_{acc} = -0.141$) had poor detection performance. Upon inspecting the images and detections, they noticed that the model had trouble identifying an object among objects of the same type. Similarly, when exploring the "boat" object data slices, they found that most of the mistakes in the boat class arose from object crowding, *e.g.*, multiple boats next to each other.

3. *Ground truth errors* were also identified. The first type of error found in this category is related to crowding errors: experts found that slices containing multiple boats often had inconsistent annotations: sometimes, a single boat was boxed, while other times, multiple boats were included in the box. This issue is also present in other objects, such as "broccoli" (multiple pieces on a plate) and "books" (multiple books on a bookshelf). A second type of error occurs when only part of an object is included in the bounding box. For example, in the chair detection, one slice corresponded to chairs where only part of the object was included in the box.

4. *Easily detected objects*. Experts also explored the data slices with the best performance, gaining additional insight into the model. For example, when exploring the "chair" object, experts found that the model performs best when the chair is large or looks like a sofa ($\{floor = 0, sofa = 1\}$, $\Delta_{acc} = 0.195$). Another interesting case comes from the "boat" object. When experts investigated the best-performing slice ($\{water = 1, boat = 1\}$, $\Delta_{acc} = 0.339$), they noticed that kayak, a particular type of boat, was easy to detect.

5. *Spurious correlations* arise due to the slice-finding method's reliance on co-occurrence to identify problematic slices. This can sometimes lead to the identification of slices that are the result of coincidental correlations rather than actual problems. A common example of this issue was found when exploring the "boat" class. The slice that had the poorest performance was characterized by the $\{clouds = 1\}$ ($\Delta_{acc} = -0.518$) itemset. However, the experts found the association of missed detection with clouds to be incorrect. Upon further investigation, they discovered that the errors were not due to the presence of clouds. Instead, the errors were the result of overcrowding issues (multiple boats in close proximity to each other) and labeling problems (boxes containing more than one boat). Similar spurious correlations happened with the slices containing sky ($\{sky = 1\}$). Despite the slice itemset indicating a false correlation between clouds and incorrect detections, the grouping of these similar errors together still allowed users to identify the mistake with relative ease.

Expert feedback. The expert users generally found the system useful for their workflow. They valued how the data slices could provide potential reasons for a model's errors, aiding them in considering various contributing factors. The simplicity of the interface was also well-received, along with the four views used to display results (original images, detection boxes, semantic segmentation, and a zoomed-in segmentation). Users also appreciated how segmentations facilitated their understanding of the model's mistakes. However, some experts noted occasional inaccuracies in the segmentation labels. While they valued the ability to alter the labels of visual concepts, they also expressed a desire to refine segments in real time, such as splitting a cluster of segments containing multiple objects. Additionally, they found data slices with concept absences occasionally difficult to comprehend, for example, the detection of the object "chair" performing poorly when no wall was present. At the end of the interview, the expert users were asked to fill out a Likert scale questionnaire about their experience with the system. Fig. 8 shows the user's responses. Overall, the responses to this questionnaire coincide with the other feedback provided.

Study Limitations. Our user study with six experts provides qualitative insights into VCR's real-world utility but has limitations in generalizability due to the small sample size. The 45-minute sessions with industry experts were challenging to scale. These findings complement our primary quantitative evaluation (Section 5.2) by illustrating how practitioners interact with the system in practice.

6. Related work

Our work draws from several research areas: interpretability frameworks for computer vision models, slice discovery methods for identifying systematic model failures, and visual concept extraction techniques. We organize the related work into these three main categories based on the core methodologies and application domains.

6.1. Explainable artificial intelligence

Our work contributes to the broader field of Explainable Artificial Intelligence (XAI), which aims to make AI systems and their results more understandable to humans [55–57]. XAI methods range from creating inherently interpretable white-box or gray-box models [58–60] to developing post-hoc explanations for black-box models through techniques like feature importance analysis [61–64] or counterfactual reasoning [65–67]. A key distinction among post-hoc explanation methods is between *local* explanations that interpret individual predictions [63,68,69] and *global* explanations that characterize overall model behavior [23,28].

VCR is most closely related to feature importance methods such as LIME [63] and SHAP [64], which explain how input features contribute to a model's outputs. VCR uses visual concepts as interpretable features and identifies correlations between these features and systematic errors through frequent itemset mining, therefore providing global explanations that characterize failure patterns across data slices. Importantly, VCR's treatment of concept absences is related to but distinct from classic feature importance methods. For example, SHAP uses additive feature attribution, decomposing a model's output into contributions from individual features based on Shapley values. A key property of this formulation is *missingness*: absent features are assumed to have zero effect on the prediction [55]. In contrast, VCR explicitly examines when the absence of a concept itself is informative about a model's failure modes.

XAI evaluation remains challenging, as criteria such as interpretability and explainability are not easily quantified. In the XAI literature, interpretability and explainability are closely related concepts that lack universal definitions [70,71]: interpretability focuses on understanding how a model arrives at its decisions (the internal mechanisms and logic), while explainability focuses on communicating why a specific decision was made in terms meaningful to users (building trust and justifying outputs). Although researchers agree that anecdotal inspection is insufficient for robust verification, the XAI community has yet to establish standardized evaluation metrics beyond often-reported anecdotal evidence showing individual, convincing examples [72–74]. VCR's evaluation addresses this challenge by combining quantitative metrics for slice discovery accuracy with qualitative evaluation through our user study, moving beyond anecdotal evidence toward more rigorous validation of explanation quality

6.2. Slice discovery methods

VCR builds upon the success of slice discovery methods in tabular datasets [2,13–17]. These methods identify problematic data subgroups using predicates over predefined attributes (*e.g.*, age = 25–40, gender = Male). DivExplorer [14], SliceLine [15], and Macrobase [16] employ optimized frequent itemset mining algorithms such as Apriori [75] and

Table 3
Feature comparison between VCR and representative frameworks for identifying systematic errors in image classification tasks.

	VCR (Ours)	SliceTeller [51]	Domino [1]	POEM [24]	ESCAPE [31]
Model agnostic	✓	✓	✓	✗	✓
Segment-level concepts	✓	✗	✗	✓	✓
Leverage metadata	✓	✓	✗	✗	✗
Automated discovery	✓	✓	✓	✓	✗
Object-detection	✓	✗	✗	✗	✗

FP-growth [43], while Slice Finder [2,13] uses decision trees and lattice search techniques. VCR uses Apriori, with optimizations specifically designed to support concept absences. When applied to image datasets, these methods rely on predefined metadata attributes. SliceTeller [51] applies frequent itemset mining to annotated attributes in datasets like CelebA [18], while Uni-Evaluator [76] handles both discrete (*e.g.*, class labels) and continuous metadata (*e.g.*, aspect ratios, size, direction) across classification, detection, and segmentation tasks. VCR extends beyond metadata-based approaches by automatically discovering visual concepts that provide additional, interpretable slicing dimensions.

A number of automated slice discovery methods seek to evaluate performance of image classification models beyond predefined metadata [1,24,32,45,77–80]. For example, Spotlight [78] identifies problematic slices of images by searching for contiguous regions in the final layer representation space of a neural network that align with errors. Domino [1] and FACTS [79] fit an error-aware Gaussian mixture model on CLIP embeddings [38] or some specialized feature space to generate data slices (clusters). Most related to us are methods that use visual concepts to explain vision model behaviors. POEM [24] uses a pre-trained semantic segmentation model on Unified Perceptual Parsing (UPP) [81] to label visual concepts and identifies a filter activation map in image classifier CNNs that overlaps with the visual concepts to use for explanations. In contrast, VCR's visual concepts are model-agnostic and not limited to predefined labels in UPP. EAC [80] uses SAM to generate segments for each image as visual concepts and uses Shapley values to explain each concept's contribution to the model's prediction, whereas VCR's notion of visual concepts focuses on the recurrences across images to capture systematic behavior. Table 3 summarizes the main features offered by representative frameworks.

6.3. Visual concept discovery

Visual concepts are conceptually related to the Bag of Visual Words (BoVW) model, which has been used to build image representations prior to deep learning methods [19–21]. The BoVW method extracts local features from images, such as via SIFT descriptors [22], and clusters these features to create a "visual vocabulary". Each image is then represented as a histogram of these visual words, analogous to how documents are represented by their constituent words and frequencies.

Modern approaches to extracting visual concepts fall into three paradigms based on supervision requirements: supervised methods require users to provide labeled examples for concepts of interest [23,27,28], automated methods that aim to extract semantically meaningful concepts without supervision [33–35], interactive approaches that allow users to collaborate with the system [29–32]. For example, ESCAPE [31] provides a workflow that allows users to select a set of semantically coherent segments to be defined as a visual concept. Similarly, VLSlice [32] is a human-in-the-loop tool that aids users in discovering data slices, but requires them to first specify the bias dimension of interest as an initial query. VCR provides automatic concept discovery by leveraging vision foundation models by default but also allows users to fine-tune the concepts in an interactive manner.

Saliency Maps and Attention Mechanisms. Saliency maps and visual concepts both aim to identify important image regions, but serve fundamentally different purposes. Saliency maps visualize which image regions contribute most to a model's prediction through gradient-based methods like GradCAM [68], GradCAM++ [82], and HiResCAM [83], or gradient-free alternatives like Eigen-CAM [84] and Ablation-CAM [85]. These pixel-level importance maps explain individual predictions by highlighting where the model "looks" for each decision. While complementary to VCR, saliency maps differ in key aspects: they explain individual predictions rather than systematic patterns, operate at the pixel level without semantic labels, and their gradient-based variants depend on specific model architectures and often require access to model internals. POEM [24] represents a hybrid approach that aligns visual concepts with model attention via saliency maps before mining patterns, but it is limited to CNN classifiers. In contrast, VCR is model-agnostic and focuses on discovering recurring semantic patterns correlated with systematic errors across data slices. In Appendix E, we provide additional visual comparisons between saliency maps and visual concepts.

7. Discussion and future work

While VCR provides interpretable concept labels and an interactive interface for refinement, it does not explicitly model individual users' familiarity with different visual concepts. Adapting explanations based on user expertise represents a promising direction for making VCR more effective across users with varying domain knowledge.

Several practical approaches could enhance the system's ability to tailor explanations to user familiarity. First, concept prioritization could boost slices containing familiar concepts higher in ranked lists, helping users quickly identify patterns in domains they understand well while still surfacing critical unfamiliar patterns when accuracy divergence is exceptionally high. Second, the concept exploration interface could automatically adjust the level of detail based on user expertise, for example, showing fine-grained concepts (*e.g.*, "sedan", "SUV", "pickup truck") in familiar domains while presenting coarser categories (*e.g.*, "vehicle") for less familiar areas. The system could also use progressive disclosure that starts with high-level familiar concepts and allows users to drill down to unfamiliar details on demand, and familiarity indicators (such as visual badges or color coding) that help users quickly distinguish between new concepts and those they have previously explored.

User familiarity could be learned implicitly from interaction patterns, such as time spent viewing concepts, which concepts are clicked versus skipped, and which concepts users merge or refine, or captured explicitly through ratings or "mark as familiar" buttons. A key challenge for these personalization mechanisms is balancing the surfacing of comfortable, familiar concepts with enabling discovery of unfamiliar but important error patterns. We leave the design and evaluation of such personalization approaches as future work.

8. Conclusion

In summary, VCR is an interactive framework for understanding systematic errors in object-detection models through visual concepts–semantically meaningful image segments that serve as interpretable primitives for slice discovery. By combining automated concept extraction with human expertise, practitioners can explore their datasets, refine concept definitions, and discover meaningful error patterns that traditional approaches miss. Through our large-scale evaluation benchmark with 1713 slice discovery settings, we show that VCR consistently outperforms alternatives in identifying problematic data slices. A user study with six industry experts provides qualitative evidence that VCR facilitates identification and explanation of errors in real-world object-detection models.

CRediT authorship contribution statement

Jie Jeff Xu: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Saahir Dhanani:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation. **Jorge Piazentin Ono:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Formal analysis, Conceptualization. **Wenbin He:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Conceptualization. **Liu Ren:** Writing – review & editing, Supervision, Project administration, Conceptualization. **Kexin Rong:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Project administration, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by the National Science Foundation under grant IIS-2335881 and by a Bosch research gift.

Appendix A. Mutual information analysis

Proposition A.1. *Consider two itemsets A and B. Assume that B is a subset of A, and that $P(A) = p, P(B) = q$. $NMI(A, B)$ decreases as $p - q$ increases.*

Proof. By definition, we have $P(A = 1, B = 1) = q, P(A = 0, B = 1) = 0, P(A = 1, B = 0) = p - q, P(A = 0, B = 0) = 1 - p$. So the mutual information for A and B is:

$$H(A) = -p\log(p) - (1-p)\log(1-p) \tag{A.1}$$

$$H(B) = -q\log(q) - (1-q)\log(1-q) \tag{A.2}$$

$$H(A,B) = -q\log(q) - (p-q)\log(p-q) - (1-p)\log(1-p) \tag{A.3}$$

$$I(A;B) = H(A) + H(B) - H(A,B) \tag{A.4}$$

$$= -p\log(p) - (1-q)\log(1-q) + (p-q)\log(p-q) \tag{A.5}$$

Let $\delta = p - q$.

$$I(A;B) = -p\log(p) - (1-(p-\delta))\log(1-(p-\delta)) + \delta\log(\delta)$$

$$\frac{\partial I(A;B)}{\partial \delta} = \log(\frac{\delta}{1-p+\delta})$$

$$H(A) + H(B) = -p\log(p) - (1-p)\log(1-p) - (p-\delta)\log(p-\delta) - (1-(p-\delta))\log(1-(p-\delta))$$

$$\frac{\partial (H(A)+H(B))}{\partial \delta} = \log(\frac{p-\delta}{1-p+\delta})$$

$NMI(A,B) = 2 \times \frac{I(A;B)}{H(A)+H(B)}$, therefore

$$\frac{\partial NMI(A,B)}{\partial \delta} = \frac{\log(\frac{\delta}{1-p+\delta})(H(A)+H(B)) + \log(\frac{p-\delta}{1-p+\delta})I(A;B)}{(H(A)+H(B))^2} \tag{A.6}$$

Plug in Eq (A.4), we also have

$$\frac{\partial NMI(A,B)}{\partial \delta} = \frac{(\log(\frac{\delta}{1-p+\delta}) + \log(\frac{p-\delta}{1-p+\delta}))(H(A)+H(B)) - \log(\frac{p-\delta}{1-p+\delta})H(A,B)}{(H(A)+H(B))^2} \tag{A.7}$$

There are two cases:

Case 1: $p - \delta < 0.5$**.** Since denominator of Eq (A.6), $H(A), H(B), I(A;B)$ are all non negative, we only need to check the sign of the numerator. When $p < 1$ and $\delta \geq 0$, $\frac{\delta}{1-p+\delta} < 1$, so $\log(\frac{\delta}{1-p+\delta}) < 0$, and the first term in the numerator is negative. When $p-\delta < 0.5$, $p-\delta < 1-(p-\delta)$, so $\log(\frac{p-\delta}{1-p+\delta}) < 0$, and the second term in the numerator is also negative. So $\frac{\partial NMI(A,B)}{\partial \delta} < 0$.

Case 2: $p - \delta \geq 0.5$**.** Similarly, we analyze the sign of the numerator of Eq. (A.7). When $p - \delta \geq 0.5 > 0$, so the second term in the numerator is negative. Let us look at the first term:

$$\log(\frac{\delta}{1-p+\delta}) + \log(\frac{p-\delta}{1-p+\delta}) = \log(\frac{\delta(p-\delta)}{(1-p+\delta)^2}) < \log(\frac{\delta}{1-p+\delta}) < 0,$$

where the first inequality follows from $p - \delta > 1 - p + \delta$, and the second inequality follows from $p < 1$. Therefore, the first term in the numerator of Eq. (A.6) is also negative. So overall $\frac{\partial NMI(A,B)}{\partial \delta} < 0$. □

Appendix B. Effect of greedy deduplication on itemset diversity

Fig. B.9 provides a qualitative example of the greedy deduplication step and quantifies its effect on result diversity. The example shows how near-duplicate itemsets (*e.g.*, {bookcase = 1, person_12 = 0} and {bookcase = 1}) are removed while preserving representative, high-divergence slices.

To assess diversity quantitatively, we compute the average pairwise Jaccard similarity among the top-50 itemsets (ranked by accuracy divergence) after applying greedy deduplication at varying maximum overlap thresholds δ. Results are from our COCO 2017 Validation dataset under 500 concept columns and 36,780 rows (bounding-box pairs). Lower Jaccard similarity indicates higher diversity.

Overall, Table B.4 shows that greedy deduplication substantially increases itemset diversity. At $\delta = 0.5$ we observe a ∼97% reduction in average pairwise overlap versus no deduplication (0.0048 vs. 0.1666). We therefore adopt $\delta = 0.5$ as the default threshold in our experiments.

Table B.4
Greedy deduplication increases diversity as the maximum allowed overlap δ decreases. At $\delta = 0.90$, average pairwise Jaccard drops by an order of magnitude compared to no deduplication ($\delta = 1.0$), and continues to fall with stricter thresholds.

δ	Average pairwise similarity
1.00	0.166626
0.95	0.023640
0.90	0.016797
0.50	0.004768
0.10	0.001252

Appendix C. Detailed description of datasets and slice settings

COCO 2014. The COCO 2014 Validation dataset is a subset of the larger Microsoft Common Objects in Context (COCO) dataset, covering a wide range of objects and scenes for a total of 40 504 images. Furthermore, each image is annotated with five descriptive captions, which we leverage in the semantic slice generation process. We chose the 2014 split over the 2017 split since our detection model was directly trained on the 2017 split.

For generating semantic slices, we focus on the top 15 object classes in the dataset, ranked by the frequency of their annotations. Specifically, among these top classes, we generate semantic slices by:

- **Filtering Images:** For each class, we filter down to images containing at least one ground truth instance of that object class.
- **Generating Image Representations:** We represent each image by embedding its captions in the CLIP embedding space. For a single image, we average the embeddings for each caption and normalize the resultant vector, denoted as v_i.

itemsets	support	supp. count	accuracy	d_accuracy
(bookcase=1, house_4=0)	0.009706	357	0.252101	-0.286643
~~(bookcase=1, person_12=0)~~	0.009788	359	0.261111	-0.277633
~~(bookcase=1)~~	0.010359	381	0.270341	-0.268403
~~(bookcase=1, wood=0)~~	0.009815	361	0.277008	-0.261736
(food_8=0, food_6=1, food=1)	0.001033	38	0.342105	-0.196639
(food_11=1, floor_6=0, food=1)	0.001196	44	0.386364	-0.152380
~~(person_42=0, wood_8=1)~~	0.001523	56	0.392857	-0.145887
~~(wood_8=1, clothes_7=0)~~	0.001523	56	0.392857	-0.145887
(wall=0, wood_8=1)	0.001523	56	0.392857	-0.145887
~~(door_3=0, wood_8=1)~~	0.001523	56	0.392857	-0.145887

itemsets	support	supp. count	accuracy
(bookcase=1, house_4=0)	0.009706	357	0.252101
(food_8=0, food_6=1, food=1)	0.001033	38	0.342105
(food_11=1, floor_6=0, food=1)	0.001196	44	0.386364
(wall=0, wood_8=1)	0.001523	56	0.392857
(bus_1=0, cupboard_2=1)	0.001006	37	0.405405
(lamp=1, house_4=0)	0.001686	62	0.419355
(person_28=0, cake=1)	0.001876	69	0.420290
(fork=0, food_15=1, food_1=1)	0.001033	38	0.421053
(food_12=1, food_1=0, food_18=1)	0.001033	38	0.421053
(bright color=1, person_12=1)	0.001223	44	0.422222

Fig. B.9. Greedy deduplication example. Left: itemsets without any deduplication. Right: itemsets after greedy deduplication using an overlap threshold $\delta = 0.5$.

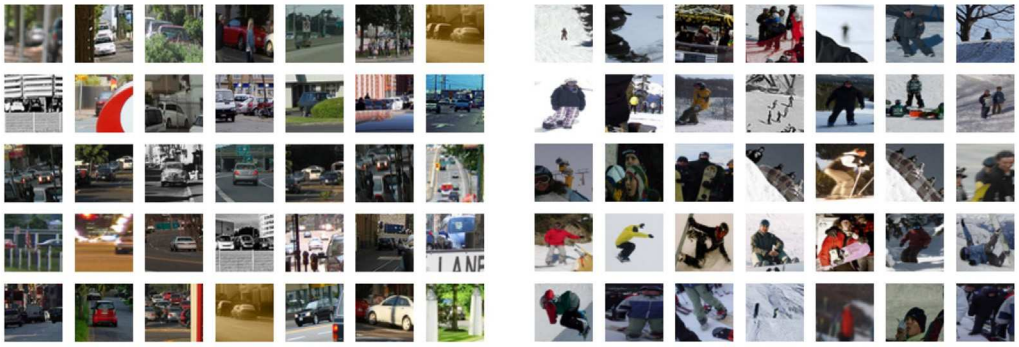


Fig. C.10. Example semantic ground truth slices generated from the COCO 2014 dataset.

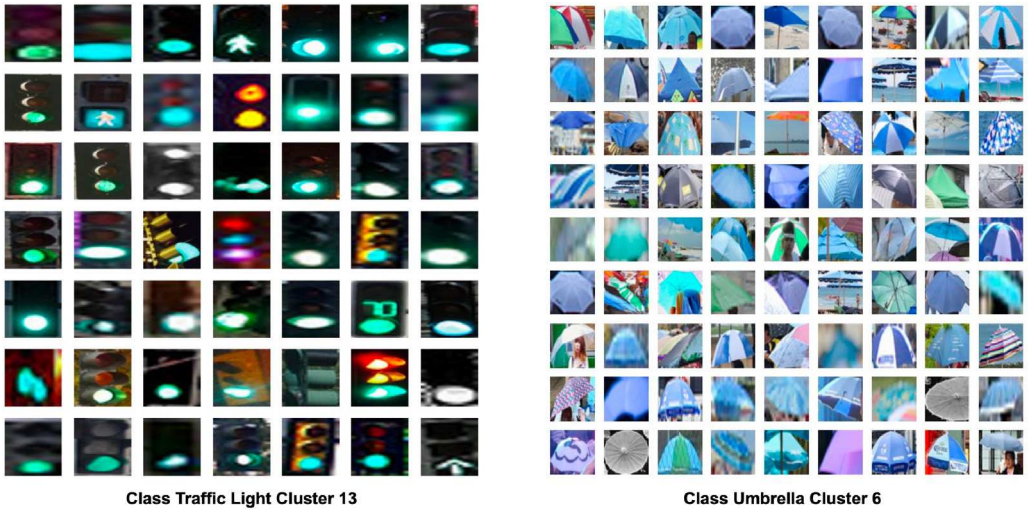


Fig. C.11. Example color slices generated from the COCO 2014 dataset. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- **Generating Contexts:** We use the template "[object class] next to [setting]" to create categories for the slices. We extract settings by considering the most frequent words in the captions of the object-filtered dataset, removing stop words (e.g., "the", "a", "of") and choosing a selection of the top nouns as settings.
- **Generating Context Representations:** We format each context string into template strings (e.g., "a photo of a big", "a photo of a small") and generate embeddings using CLIP. By using these templates, we can generate a variety of contexts that describe the object class in different ways, such as its size, quality, or appearance. We average the CLIP embeddings and normalize the resultant vector, denoted as c_j, representing one context.
- **Finding Closest Semantic Slice:** To categorize the image representations, we find the context vector, c_j, with the smallest distance to the image vector, v_i, indicating the closest semantic slice. Once the closest semantic slice is determined, every instance of the object class in that image is assigned to that category.

Fig. C.10 shows a few examples of semantic ground truth slices generated via this approach. For the three metadata-based settings, we again use the 15 classes with the most number of labels. For

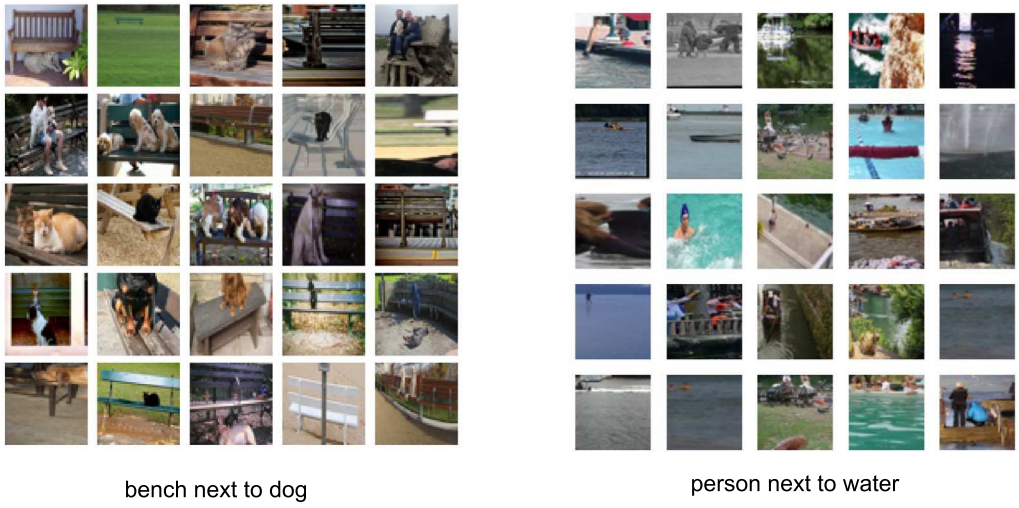


Fig. C.12. Example semantic ground truth slices generated from the Visual Genome dataset.

both size and aspect ratio bounding box metadata settings, we bucket bounding boxes into 5–15 different bins according to their size/aspect ratio. For color clusters, we extract 50 × 50 square crops of ground-truth bounding boxes and run them through the K-Means clustering algorithm for k = 20 total clusters. Fig. C.11 shows examples of color slices generated for this dataset.

Visual Genome. The Visual Genome (VG) dataset is a large-scale image dataset covering a wide range of everyday scenes and objects, with more than 108,000 images, each annotated with dense object annotations, attributes, and relationship graphs. Instead of using all 108,000 images, we start with the first half of the dataset VG_100K (where the second half is VG_100K_2) and filter it down the images further to include only images that have any of the 80 COCO labels. Additionally, VG is known to have image overlaps with COCO. As such, we ignore any of the images that have an associated COCO "id" with it.

In order to generate semantic slices we perform the following steps:

- **Filtering Images:** For each class, we filter down to images containing at least one ground truth instance of that object class.
- **Image Representation:** We generate a set of captions for each image by selecting a diverse set of regions within the image. This selection includes:
  - A subset of the largest regions by bounding box area, representing the most prominent objects.
  - Middle-sized regions, providing contextual information and additional details.
  - A random sample of the smallest regions, offering diversity and capturing less prominent elements.

  We concatenate the phrases from these regions in groups of three to generate a set of captions for the image. This helps to form captions that encapsulate the image's coarse and fine details. We embed these captions in the CLIP embedding space and take the average of the embeddings to represent the image. This average serves as the image representation v_i.
- **Generating Contexts:** We use the text template "[object class] next to [settings]" to generate contexts for the slice categories. We find the settings by selecting from the most common nouns found in the phrases associated with each region in an image after filtering out stop words.
- **Context Representation:** We format each context into template strings (e.g., "a photo of", "a photo of a small") to generate a variety of contexts that describe the object class in different ways, such as its size, quality, or appearance. We generate embeddings using CLIP for each context template, average the CLIP embeddings, and normalize the resultant vector, denoted as c_j, representing one context.
- **Finding Closest Semantic Slice:** To categorize the image representations, we find the context vector c_j with the smallest distance to the image vector v_i, indicating the closest semantic slice. This is done by calculating the L2 distance $\|v_i - c_j\|^2$ and selecting the context vector c_j that minimizes this distance. Once the closest semantic slice is determined, every instance of the object class in that image is assigned to that category.

We display a few of these semantic slices in Fig. C.12. For the three metadata based settings, we do the same as with the COCO dataset, bucketing the different sizes and aspect ratios for the bounding boxes, and using K-Means to form color clusters.

BDD100K. The BDD100K dataset consists of 100,000 images taken from the perspective of a car, featuring diverse scenes across various times and conditions. Unlike COCO and VGG, BDD100K has a very limited number of classes, most of which overlap with COCO's. To make sure BDD100K's detection annotations align with those of our object detector, we remove the "traffic sign" label and merge "rider" and "pedestrian" into "person". We further created our own 10K split of BDD100K after filtering for images that contain object-detections after finding that the provided BDD10K did not always have detection labels.

From this subset of BDD data, we then created semantic slices based directly on metadata provided by BDD (no CLIP needed). Specifically, we used the following metadata attributes to create semantic slices:

- timeofday: daytime, night, dawn, dusk, undefined
- weather: rainy, snowy, clear, overcast, partly cloudy, foggy, undefined
- scene: tunnel, residential, parking lot, city street, gas stations, highway, undefined

This leads to slices of the form "[object class] in [metadata attribute]". We display a few of these semantic slices in Fig. C.13.

For the three metadata based settings, we do the same as with the COCO dataset, bucketing the different sizes and aspect ratios for the bounding boxes, and using K-Means to form color clusters.

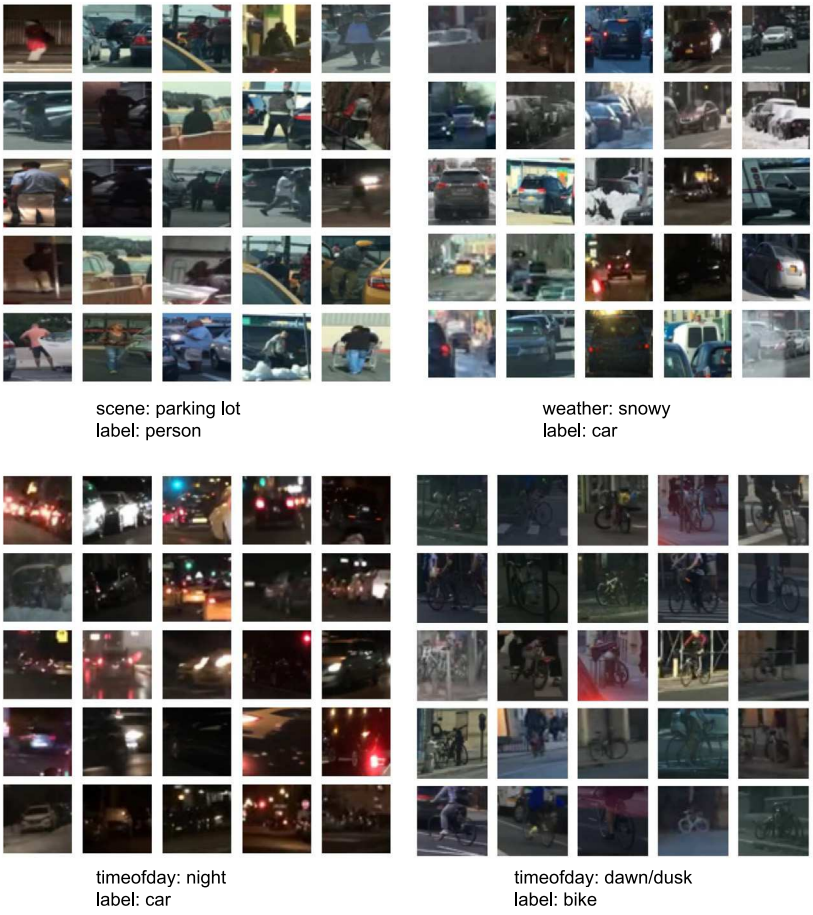


Fig. C.13. Example semantic ground truth slices generated from the BDD dataset. The first line under each image represents the semantic setting, the second line represents the target class.

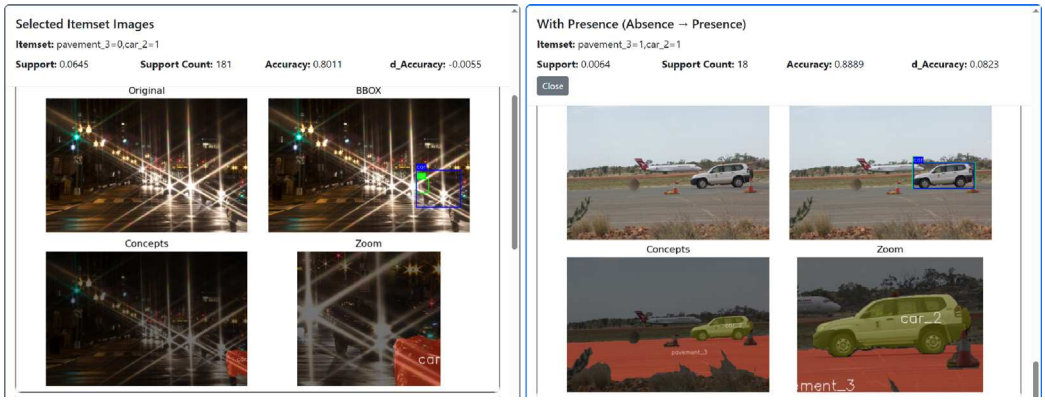


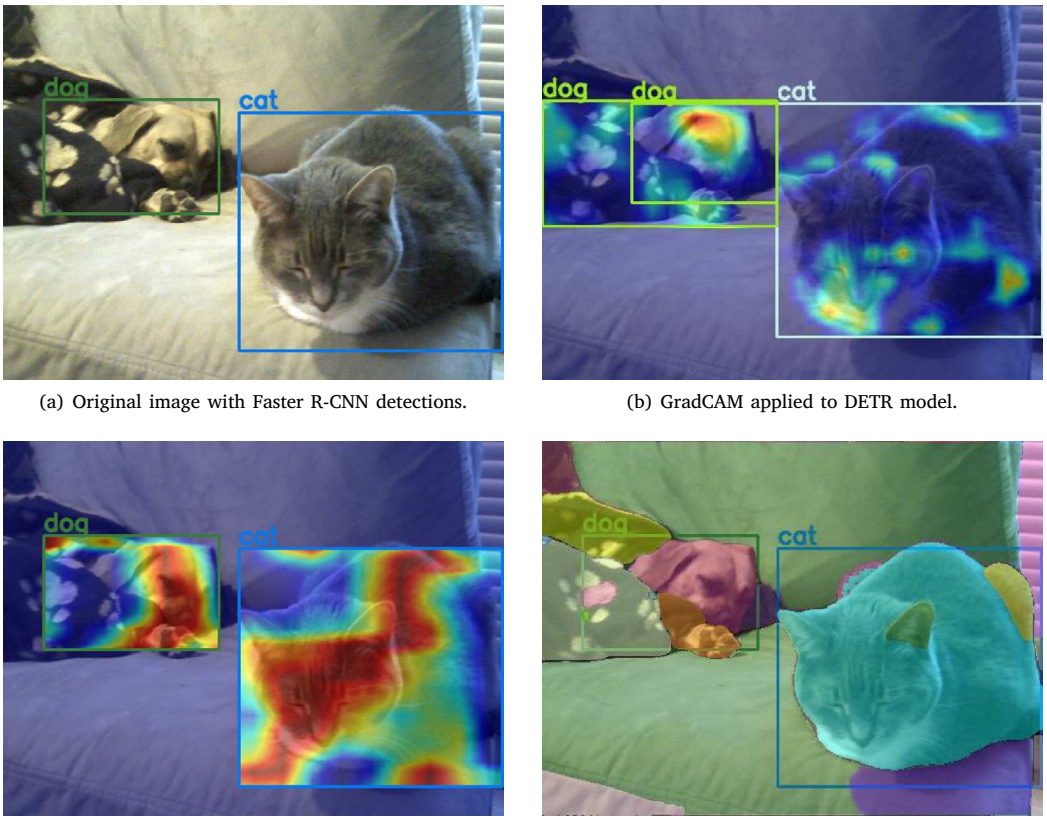
Fig. D.14. Visualization panel for comparing concept absences versus presences. The left panel shows an itemset with concept absence, {pavement_3 = 0, car_2 = 1}, where glare from car lights introduces visual noise and obscures the roadway, resulting in poor IoU. The right panel displays the alternative scenario, {pavement_3 = 1, car_2 = 1}, where the road is clearly present and the model achieves high IoU for the car.

Appendix D. Visualization of concept absence

To help users interpret the effects of concept absences, we introduce a comparison-based visualization mechanism to our interface as seen in Fig. D.14. Specifically, when a concept is absent, users can now view an alternative visualization in which that concept is present, while keeping the rest of the itemset unchanged. This side-by-side presentation allows users to better contextualize how the presence or absence of specific concepts influences the overall interpretation.

Appendix E. Comparison with saliency maps

While saliency maps provide pixel-level importance for individual predictions, VCR visual concepts offer semantic, segment-level representations that recur across images to identify systematic error patterns.



(a) Original image with Faster R-CNN detections.

(b) GradCAM applied to DETR model.

(c) Eigen-CAM applied to Faster R-CNN.

(d) SAM segmentations as visual concepts.

Fig. E.15. Comparison between saliency maps and visual concepts. Saliency maps (b, c) provide pixel-level attention for individual predictions without semantic labels, while VCR's visual concepts (d) produce semantically-labeled segments that can be mined across images to identify systematic error patterns.

Fig. E.15 illustrates this distinction using the same input image. The original image (a) shows the ground truth bounding boxes detected by the model. Both GradCAM applied to DETR [86] (b) and Eigen-CAM applied to Faster R-CNN (c) highlight pixel-level regions of importance for individual predictions, but these heatmaps are prediction-specific and lack semantic labels. Note that different saliency methods are required for different architectures: DETR's transformer-based architecture supports gradient-based methods like GradCAM, while Faster R-CNN requires gradient-free alternatives like Eigen-CAM. In contrast, our SAM-based segmentation approach (d) produces interpretable, semantically-coherent segments (e.g., "car", "road", "sky") that serve as visual concepts. These concepts can be mined across images to discover systematic failure patterns, whereas saliency maps explain only individual decisions without revealing recurring error correlations.

Data availability

Data will be made available on request.

References

[1] S. Eyuboglu, M. Varma, K. Saab, J.-B. Delbrouck, C. Lee-Messer, J. Dunnmon, J. Zou, C. Ré, Domino: Discovering systematic errors with cross-modal embeddings, 2022, arXiv preprint arXiv:2203.14960.

[2] Y. Chung, T. Kraska, N. Polyzotis, K.H. Tae, S.E. Whang, Slice finder: Automated data slicing for model validation, in: 2019 IEEE 35th International Conference on Data Engineering, ICDE, IEEE, 2019, pp. 1550–1553.

[3] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, ACM Comput. Surv. 54 (6) (2021) 1–35.

[4] T. De Vries, I. Misra, C. Wang, L. Van der Maaten, Does object recognition work for everyone? in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019, pp. 52–59.

[5] K. Xiao, L. Engstrom, A. Ilyas, A. Madry, Noise or signal: The role of image backgrounds in object recognition, 2020, arXiv preprint arXiv:2006.09994.

[6] A. Barbu, D. Mayo, J. Alverio, W. Luo, C. Wang, D. Gutfreund, J. Tenenbaum, B. Katz, Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models, Adv. Neural Inf. Process. Syst. 32 (2019).

[7] J. Buolamwini, T. Gebru, Gender shades: Intersectional accuracy disparities in commercial gender classification, in: Conference on Fairness, Accountability and Transparency, PMLR, 2018, pp. 77–91.

[8] R. Shetty, B. Schiele, M. Fritz, Not using the car to see the sidewalk–quantifying and controlling the effects of context in classification and segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8218–8226.

[9] A.J. DeGrave, J.D. Janizek, S.-I. Lee, AI for radiographic COVID-19 detection selects shortcuts over signal, Nat. Mach. Intell. 3 (7) (2021) 610–619.

[10] A. Bissoto, E. Valle, S. Avila, Debiasing skin lesion datasets and models? not so fast, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 740–741.

[11] R. Zellers, Y. Bisk, R. Schwartz, Y. Choi, Swag: A large-scale adversarial dataset for grounded commonsense inference, 2018, arXiv preprint arXiv:1808.05326.

[12] K. Karkkainen, J. Joo, Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 1548–1558.

[13] Y. Chung, T. Kraska, N. Polyzotis, K.H. Tae, S.E. Whang, Automated data slicing for model validation: A big data-ai integration approach, IEEE Trans. Knowl. Data Eng. 32 (12) (2019) 2284–2296.

[14] E. Pastor, L. De Alfaro, E. Baralis, Looking for trouble: Analyzing classifier behavior via pattern divergence, in: Proceedings of the 2021 International Conference on Management of Data, 2021, pp. 1400–1412.

[15] S. Sagadeeva, M. Boehm, Sliceline: Fast, linear-algebra-based slice finding for ml model debugging, in: Proceedings of the 2021 International Conference on Management of Data, 2021, pp. 2290–2299.

[16] P. Bailis, E. Gan, S. Madden, D. Narayanan, K. Rong, S. Suri, Macrobase: Prioritizing attention in fast data, in: Proceedings of the 2017 ACM International Conference on Management of Data, 2017, pp. 541–556.

[17] E. Pastor, E. Baralis, L. de Alfaro, et al., A hierarchical approach to anomalous subgroup discovery, in: 39th IEEE International Conference on Data Engineering, ICDE 2023, Anaheim, California, USA, April 3–7, 2023., IEEE, 2023.

[18] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3730–3738.

[19] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Workshop on Statistical Learning in Computer Vision, ECCV, vol. 1, Prague, 2004, pp. 1–2.

[20] F. Perronnin, C. Dance, Fisher kernels on visual vocabularies for image categorization, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–8.

[21] L. Fei-Fei, P. Perona, A bayesian hierarchical model for learning natural scene categories, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'05, vol. 2, IEEE, 2005, pp. 524–531.

[22] D.G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, Ieee, 1999, pp. 1150–1157.

[23] D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, Network dissection: Quantifying interpretability of deep visual representations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6541–6549.

[24] V. Dadvar, L. Golab, D. Srivastava, POEM: Pattern-oriented explanations of convolutional neural networks, Proc. VLDB Endow. 16 (11) (2023) 3192–3200.

[25] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A.C. Berg, W.-Y. Lo, et al., Segment anything, 2023, arXiv preprint arXiv:2304.02643.

[26] X. Dong, J. Bao, Y. Zheng, T. Zhang, D. Chen, H. Yang, M. Zeng, W. Zhang, L. Yuan, D. Chen, et al., Maskclip: Masked self-distillation advances contrastive language-image pretraining, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 10995–11005.

[27] B. Zhou, Y. Sun, D. Bau, A. Torralba, Interpretable basis decomposition for visual explanation, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 119–134.

[28] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, et al., Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav), in: International Conference on Machine Learning, PMLR, 2018, pp. 2668–2677.

[29] Z. Zhao, P. Xu, C. Scheidegger, L. Ren, Human-in-the-loop extraction of interpretable concepts in deep learning models, IEEE Trans. Vis. Comput. Graphics 28 (1) (2021) 780–790.

[30] J. Huang, A. Mishra, B.C. Kwon, C. Bryan, ConceptExplainer: Interactive explanation for deep neural networks from a concept perspective, IEEE Trans. Vis. Comput. Graphics 29 (1) (2022) 831–841.

[31] Y. Ahn, Y.-R. Lin, P. Xu, Z. Dai, ESCAPE: Countering systematic errors from machine's blind spots via interactive visual analysis, in: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI '23, Association for Computing Machinery, New York, NY, USA, 2023, http://dx.doi.org/10.1145/3544548.3581373.

[32] E. Slyman, M. Kahng, S. Lee, VLSlice: Interactive vision-and-language slice discovery, in: International Conference on Computer Vision, ICCV, 2023, URL https://arxiv.org/pdf/2309.06703.pdf.

[33] A. Ghorbani, J. Wexler, J.Y. Zou, B. Kim, Towards automatic concept-based explanations, Adv. Neural Inf. Process. Syst. 32 (2019).

[34] H. Park, N. Das, R. Duggal, A.P. Wright, O. Shaikh, F. Hohman, D.H.P. Chau, Neurocartography: Scalable automatic visual summarization of concepts in deep neural networks, IEEE Trans. Vis. Comput. Graphics 28 (1) (2021) 813–823.

[35] M. Hoque, W. He, A. Shekar, L. Gou, L. Ren, Visual concept programming: A visual analytics approach to injecting human intelligence at scale, IEEE Trans. Vis. Comput. Graphics 29 (01) (2023) 74–83, http://dx.doi.org/10.1109/TVCG.2022.3209466.

[36] J. Vogel, B. Schiele, Semantic modeling of natural scenes for content-based image retrieval, Int. J. Comput. Vis. 72 (2007) 133–157.

[37] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, 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, 2014, pp. 740–755.

[38] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, PMLR, 2021, pp. 8748–8763.

[39] F. Liang, B. Wu, X. Dai, K. Li, Y. Zhao, H. Zhang, P. Zhang, P. Vajda, D. Marculescu, Open-vocabulary semantic segmentation with mask-adapted clip, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 7061–7070.

[40] D. Bolya, S. Foley, J. Hays, J. Hoffman, 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, 2020, pp. 558–573.

[41] L. McInnes, J. Healy, J. Melville, Umap: Uniform manifold approximation and projection for dimension reduction, 2018, arXiv preprint arXiv:1802.03426.

[42] J.J. Xu, S. Dhanani, J.P. Ono, W. He, L. Ren, K. Rong, Demonstration of VCR: A tabular data slicing approach to understanding object detection model performance, Proc. VLDB Endow. 17 (12) (2024) 4453–4456.

[43] J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, ACM Sigmod Rec. 29 (2) (2000) 1–12.

[44] D. Hoiem, Y. Chodpathumwan, Q. Dai, Diagnosing error in object detectors, in: European Conference on Computer Vision, Springer, 2012, pp. 340–353.

[45] G. Plumb, N. Johnson, A. Cabrera, A. Talwalkar, Towards a more rigorous science of blindspot discovery in image classification models, Trans. Mach. Learn. Res. (2023).

[46] S. Joshi, Y. Yang, Y. Xue, W. Yang, B. Mirzasoleiman, Towards mitigating spurious correlations in the wild: A benchmark & a more realistic dataset, 2023, arXiv preprint arXiv:2306.11957.

[47] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D.A. Shamma, et al., Visual genome: Connecting language and vision using crowdsourced dense image annotations, Int. J. Comput. Vis. 123 (2017) 32–73.

[48] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, T. Darrell, Bdd100k: A diverse driving dataset for heterogeneous multitask learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2636–2645.

[49] MMDetection Github, 2024, https://github.com/open-mmlab/mmdetection. (Accessed Februrary 2024).

[50] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, et al., MMDetection: Open mmlab detection toolbox and benchmark, 2019, arXiv preprint arXiv:1906.07155.

[51] X. Zhang, J.P. Ono, H. Song, L. Gou, K.-L. Ma, L. Ren, SliceTeller: A data slice-driven approach for machine learning model validation, IEEE Trans. Vis. Comput. Graphics 29 (1) (2022) 842–852.

[52] Seeing in concepts: Enabling structured image representation and analysis with visual concepts (technical report), 2024, https://anonymous.4open.science/r/seeing-in-concepts-4495/docs/tr.pdf. (Accessed October 2024).

[53] DivExplorer github, 2024, https://github.com/divexplorer/divexplorer. (Accessed Februrary 2024).

[54] SliceLine github, 2024, https://github.com/DataDome/sliceline. (Accessed Februrary 2024).

[55] E.S. Ortigossa, T. Gonçalves, L.G. Nonato, Explainable artificial intelligence (xai)—from theory to methods and applications, IEEE Access 12 (2024) 80799–80846.

[56] S. Ali, T. Abuhmed, S. El-Sappagh, K. Muhammad, J.M. Alonso-Moral, R. Confalonieri, R. Guidotti, J. Del Ser, N. Díaz-Rodríguez, F. Herrera, Explainable artificial intelligence (XAI): What we know and what is left to attain trustworthy artificial intelligence, Inf. Fusion 99 (2023) 101805.

[57] M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M. Van Keulen, C. Seifert, From anecdotal evidence to quantitative evaluation methods: A systematic review on evaluating explainable ai, ACM Comput. Surv. 55 (13s) (2023) 1–42.

[58] G.P. Schmitz, C. Aldrich, F.S. Gouws, ANN-DT: An algorithm for extraction of decision trees from artificial neural networks, IEEE Trans. Neural Netw. 10 (6) (1999) 1392–1401.

[59] Y. Lou, R. Caruana, J. Gehrke, G. Hooker, Accurate intelligible models with pairwise interactions, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2013, pp. 623–631.

[60] N. Papernot, P. McDaniel, Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning, 2018, arXiv preprint arXiv:1803.04765.

[61] G. Casalicchio, C. Molnar, B. Bischl, Visualizing the feature importance for black box models, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2018, pp. 655–670.

[62] M. Wojtas, K. Chen, Feature importance ranking for deep learning, Adv. Neural Inf. Process. Syst. 33 (2020) 5105–5114.

[63] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should i trust you?" explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.

[64] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Adv. Neural Inf. Process. Syst. 30 (2017).

[65] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the GDPR, Harv. JL & Tech. 31 (2017) 841.

[66] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, F. Giannotti, Local rule-based explanations of black box decision systems, 2018, arXiv preprint arXiv:1805.10820.

[67] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. De Bie, P. Flach, FACE: Feasible and actionable counterfactual explanations, in: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, 2020, pp. 344–350.

[68] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.

[69] V. Petsiuk, A. Das, K. Saenko, Rise: Randomized input sampling for explanation of black-box models, 2018, arXiv preprint arXiv:1806.07421.
[70] F. Doshi-Velez, B. Kim, Towards a rigorous science of interpretable machine learning, 2017, arXiv preprint arXiv:1702.08608.
[71] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, G.-Z. Yang, XAI—Explainable artificial intelligence, Sci. Robot. 4 (37) (2019) eaay7120.
[72] M.L. Leavitt, A. Morcos, Towards falsifiable interpretability research, 2020, arXiv preprint arXiv:2010.12016.
[73] F. Doshi-Velez, B. Kim, Considerations for evaluation and generalization in interpretable machine learning, in: Explainable and Interpretable Models in Computer Vision and Machine Learning, Springer, 2018, pp. 3–17.
[74] A. Jacovi, Y. Goldberg, Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness?, 2020, arXiv preprint arXiv:2004.03685.
[75] R. Agrawal, R. Srikant, et al., Fast algorithms for mining association rules, in: Proc. 20th Int. Conf. Very Large Data Bases, VLDB, vol. 1215, Santiago, Chile, 1994, pp. 487–499.
[76] C. Chen, Y. Guo, F. Tian, S. Liu, W. Yang, Z. Wang, J. Wu, H. Su, H. Pfister, S. Liu, A unified interactive model evaluation for classification, object detection, and instance segmentation in computer vision, IEEE Trans. Vis. Comput. Graphics 30 (1) (2024) 76–86, http://dx.doi.org/10.1109/TVCG.2023.3326588.
[77] N. Sohoni, J. Dunnmon, G. Angus, A. Gu, C. Ré, No subclass left behind: Fine-grained robustness in coarse-grained classification problems, Adv. Neural Inf. Process. Syst. 33 (2020) 19339–19352.
[78] G. d'Eon, J. d'Eon, J.R. Wright, K. Leyton-Brown, 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, 2022, pp. 1962–1981.
[79] S. Yenamandra, P. Ramesh, V. Prabhu, J. Hoffman, FACTS: First amplify correlations and then slice to discover bias, in: IEEE/CVF International Conference in Computer Vision, ICCV, 2023.
[80] A. Sun, P. Ma, Y. Yuan, S. Wang, Explain any concept: Segment anything meets concept-based explanation, 2023, arXiv preprint arXiv:2305.10289.
[81] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, J. Sun, Unified perceptual parsing for scene understanding, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 418–434.
[82] A. Chattopadhay, A. Sarkar, P. Howlader, V.N. Balasubramanian, Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks, in: 2018 IEEE Winter Conference on Applications of Computer Vision, WACV, IEEE, 2018, pp. 839–847.
[83] R.L. Draelos, L. Carin, Use HiResCAM instead of grad-CAM for faithful explanations of convolutional neural networks, 2020, arXiv preprint arXiv:2011.08891.
[84] M.B. Muhammad, M. Yeasin, Eigen-cam: Class activation map using principal components, in: 2020 International Joint Conference on Neural Networks, IJCNN, IEEE, 2020, pp. 1–7.
[85] H.G. Ramaswamy, et al., Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 983–991.
[86] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable detr: Deformable transformers for end-to-end object detection, 2020, arXiv preprint arXiv:2010.04159.