Eurographics Conference on Visualization (EuroVis) 2025
W. Aigner, N. Andrienko, and B. Wang
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 44 (2025), Number 3

VISLIX: An XAI Framework for Validating Vision Models with Slice Discovery and Analysis

Xinyuan Yan[1], Xiwei Xuan[2], Jorge Piazentin Ono[3], Jiajing Guo[3], Vikram Mohanty[3], Shekar Arvind Kumar[4], Liang Gou[5], Bei Wang[1], Liu Ren[3]

[1]Scientific Computing and Imaging Institute, University of Utah, USA
[2]University of California, Davis, USA
[3]Bosch Research North America and Bosch Center for Artificial Intelligence (BCAI), USA
[4]Robert Bosch GmbH, Germany
[5]Splunk Technology, USA

Abstract

Real-world machine learning models require rigorous evaluation before deployment, especially in safety-critical domains like autonomous driving and surveillance. The evaluation of machine learning models often focuses on data slices, which are subsets of the data that share a set of characteristics. Data slice finding automatically identifies conditions or data subgroups where models underperform, aiding developers in mitigating performance issues. Despite its popularity and effectiveness, data slicing for vision model validation faces several challenges. First, data slicing often needs additional image metadata or visual concepts, and falls short in certain computer vision tasks, such as object detection. Second, understanding data slices is a labor-intensive and mentally demanding process that heavily relies on the expert's domain knowledge. Third, data slicing lacks a human-in-the-loop solution that allows experts to form hypothesis and test them interactively. To overcome these limitations and better support the machine learning operations lifecycle, we introduce VISLIX*, a novel visual analytics framework that employs state-of-the-art foundation models to help domain experts analyze slices in computer vision models. Our approach does not require image metadata or visual concepts, automatically generates natural language insights, and allows users to test data slice hypothesis interactively. We evaluate* VISLIX *with an expert study and three use cases, that demonstrate the effectiveness of our tool in providing comprehensive insights for validating object detection models.*

CCS Concepts

• ***Computing methodologies*** → *Model verification and validation; Computer vision tasks;* • ***Human-centered computing*** → *Visual analytics; Interactive systems and tools;*

1. Introduction

Computer vision models are widely employed across various domains, including autonomous driving, where they detect nearby objects [CBL*20, GZL*20], and surveillance, where they identify suspicious activities [ŞSAAH*23]. Although computer vision models often achieve high overall performance, they may underperform on semantically coherent subsets of data, known as *data slices* or *edge cases* [ZOS*22, SB21]. For example, adverse weather can hinder car detection [ZCYT23], and skin tones may affect pedestrian detection across demographic groups [WHM19]. To ensure safety, robustness, and fairness, AI developers must efficiently identify, understand, and address such slices before deployment [RCO*19].

Data slicing, a popular model validation technique, automatically identifies visually consistent yet underperforming data subgroups. It often relies on image metadata—textual labels for image attributes (e.g., weather = "rainy", lighting = "dark")—to partition subgroups [BSI*24, ZOS*22], which, however, is labor-intensive to acquire. To reduce this burden, *ConceptSlicer* [ZPOH*24] automatically checks the presence of predefined visual concepts (e.g., "bus = 1", "car = 0") using semantic segmentation models. However, both image metadata and segment tags can miss important image details. More advanced methods [EVS*22, ddWLB22] use clustering-based techniques on image embeddings to identify error-consistent groups. However, these approaches are designed for image classifiers that consider global context and are not well-suited to object detectors, which prioritize local context [BHM23].

After identifying slices, experts must analyze them to form hypothesis about the model's failure modes. This interpretation step is essential for downstream tasks like model optimization [EVS*22] and stakeholder communication [BRYB23]. Yet, the process is demanding, requiring experts to examine slice images, infer failure root causes, and summarize common patterns. A recent study [JCPT23] further highlights the complexity of this task, as experts may hold inconsistent or biased explanations. Besides ana-

lyzing the data slices, experts often come with their own hypothesis about model failures. However, existing slice discovery methods are meant to run only once and cannot interact with users [JCPT23].

To address these challenges, we present *VISLIX* (VIsual SLIce eXplanations), an explainable artificial intelligence (XAI) framework that leverages Large Language Models (LLMs) and Vision-Language Models (VLMs) to support human-in-the-loop slice discovery and analysis for vision model validation, with a focus on object detection. *VISLIX* identifies slices using context-aware image embeddings to uncover systematic errors. It then generates natural language data slice explanations, using a conversational vision-language framework. Our visual analytics system enables experts to efficiently explore slices, inspect and refine slice details, and test hypothetical scenarios through direct visual manipulation and natural language queries. In summary, our framework makes the following contributions:

- We propose a new slice discovery method tailored for object detector validation, which does not rely on either image metadata or visual concepts. Our approach is able to automatically describe and explain problematic data slices in natural language.
- We design a visual analytics system that enables users to efficiently explore data slices and interactively test new hypotheses by creating slices via both visual and textual queries.
- We validate *VISLIX* with an expert study and three use cases, which demonstrate that *VISLIX* efficiently finds and explains a broad spectrum of data slices. We also show how insights gained from *VISLIX* can be leveraged to enhance model robustness through fine-tuning.

2. Related Work

Data slice finding. Slice-finding techniques uncover data subgroups where ML models underperform, exposing systematic errors [CKP*19, PBdA*23]. Tools like *DivExplorer* [PDAB21], *SliceLine* [SB21], and *Macrobase* [BGM*17] use frequent itemset mining (FIM) algorithms for slice discovery, such as Apriori [AS*94] and FP-growth [HPY00]. These approaches have been adapted for image datasets by using metadata attributes to define subgroups [ZOS*22, CGT*24]. However, obtaining structured metadata for images remains challenging [XOG*25, XDLM24].

ConceptSlicer [ZPOH*24] employs an image segmentation model to identify predefined concepts (e.g., *person*, *bus*, *car*) and treat them as metadata, analyzing performance of concept combinations (e.g., $bus = 1$ & $car = 0$). Yet, its reliance on predefined concepts limits its descriptiveness and coverage. To address this issue, many methods encode images into latent spaces and cluster them to find problematic slices [WAG22, EVS*22, ddWLB22], e.g., *GEORGE* [SDA*20] and *UDIS* [KPSH21] use over-clustering and hierarchical clustering, whereas *AttributionScanner* [XOG*25] applies K-means on attribution-weighted embeddings to find spurious correlation slices. However, these methods target image classifiers that utilize the entire image and are not well-suited for object detection, where the object and its surroundings are crucial [LOW*20].

To capture local context, prior work has expanded the detection window and fused their embeddings during model training to enhance performance [LWL*16, ZZW*17]. We adapt this approach to a different setting: post hoc slice discovery, enabling more effective validation for object detection tasks.

Data slice explainability. Prior research on interpreting data slices follows two main approaches. The first utilizes saliency maps [LWHC22, XDL*24] to highlight image regions influencing model predictions. While effective for individual images, this method requires domain expertise [BRYB23], risks confirmation bias [AGM*18], and lacks scalability. In contrast, natural language explanations have gained traction for their clarity and utility in tasks such as stakeholder communication [BRYB23], failure reporting [CDHP21], and model fine-tuning [ZPOH*24]. The rise of foundation models, including LLMs like ChatGPT [Ope22] and LLaMA 2 [TMS*23], and VLMs like CLIP [RKH*21], BLIP-2 [LLSH23], LLaVA [LLWL24, LLLL24], and GPT-4 [Ope23], has enabled more sophisticated textual slice explanations. For example, *Domino* [EVS*22] uses CLIP to describe slices via predefined templates, while Jain et al. [JLMM22] and Wiles et al. [WAG22] leverage captioning models to summarize slice content. *ConceptSlicer* [ZOS*22] applies LLMs to describe general scenes based on visual concepts, primarily for data augmentation. However, these methods often provide high-level descriptions, overlooking nuanced errors. Our approach leverages generative VLMs and LLMs to produce template-free summaries of slice errors, capturing subtle details and significantly expanding the scope of explanations.

Interactive systems for data slice analysis. Various interactive systems support slice exploration for vision model validation [ZOS*22, KPSH21, CDHP21, CFB*23, LWHC22, XOG*25, PDAB21]. Some systems first generate slices and present them in an overview+detail format. For instance, *SliceTeller* [ZOS*22] uses *DivExplorer* [PDAB21] to create slices and visualizes them with matrix encodings, with details in a separate panel. *VISCUIT* [LWHC22] lists slices generated by *UDIS* [KPSH21] and enables neuron activation inspection. *ConceptSlicer* [ZPOH*24] supports slice viewing, concept inspection, and training data augmentation, while *Uni-Evaluator* [CGT*24] visualizes global performance and slices using matrix, table, and grid-based encodings. Interactive slice discovery tools include *AdaVision* [GILR23], which retrieves images via natural language for iterative testing, and *ESCAPE* [ALXD23], which identifies spurious associations in image classifiers with UMAP plots. Our system combines efficient slice inspection with interactive discovery, leveraging visual exploration and natural language queries.

3. Background on Object Detection

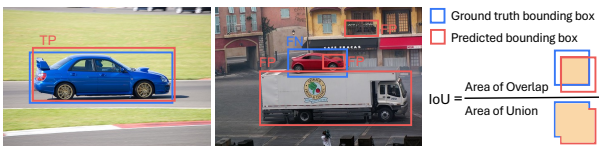

Figure 1: *Illustrations of True Positive (TP), False Positive (FP), and False Negative (FN) for a car detector and IoU computation. Left: $IoU > 0.5$ (correct). Middle: $0 \leq IoU < 0.5$ (incorrect).*

Object detection identifies object classes (e.g., car, pedestrian) and their spatial locations (e.g., bounding boxes) in an image. Given an object detector like a car detector, the goodness of the

predicted car location is accessed via the *Intersection over Union* (IoU), a metric computed as the ratio of the intersection area to the union area of predicted and ground truth bounding boxes. A detection is considered correct if IoU exceeds a threshold, typically set to 0.5. With this threshold, a bounding box is labeled as one of three types, as shown in Fig. 1: *True Positive* (TP), where the detector correctly identifies a car; *False Positive* (FP), where the detector erroneously identifies other objects as a car; *False Negative* (FN), where the ground truth car is not correctly detected. The detection outcome is influenced by both object characteristics like color and pose, and its surrounding context that offers valuable cues for model perception [LOW*20, ZCS*23]. For example, the unidentified red car in Fig. 1 (middle) could be attributed to its unusual position on top of a truck. To quantify the overall performance of a detector, the most widely used metric is *mean Average Precision* (mAP) derived from precision (i.e., $\frac{TPs}{TPs+FPs}$) and recall (i.e., $\frac{TPs}{TPs+FNs}$). See [ZCS*23] for a complete survey.

4. Domain Challenges and Design Requirements

Throughout this project, we closely collaborated with two industry computer vision experts to understand their needs and gather insights. Both experts hold PhDs in machine learning and work on various computer vision tasks for autonomous driving and driver assistance systems. Based on their feedback and an extensive literature review, this section outlines the key challenges (Sec. 4.1) that domain experts face in identifying and understanding data slices, particularly for object detector validation. These challenges motivate the formulation of our design requirements (Sec. 4.2).

4.1. Domain Challenges

C1. Existing slice discovery methods fall short in object detection. Prior methods often rely on expert-defined image metadata [IBB*22] or visual concepts [ZPOH*24]. However, anticipating all critical slices is impractical [CKP*19], and such data misses complex scenarios (e.g., "car occluded by a tree"). Recent methods [EVS*22, ddWLB22] use image embeddings and clustering to generate slices. While effective for image classifiers, they struggle with object detectors, as global embeddings overlook detection locality and cannot distinguish multiple detections in the same image.

C2. Interpreting data slices is a nontrivial task. Experts interpret slices to uncover model failure patterns, guiding refinement and deployment decisions [CDHP21]. This process—examining slice images, inferring failure causes, and summarizing patterns—is time-consuming and prone to bias or errors [JCPT23]. To simplify this task, prior methods have used predefined templates [EVS*22] or sampled image captions [WAG22, JLMM22], but these approaches lack flexibility and fail to adequately explain detection errors.

C3. A human-in-the-loop solution is needed for slice exploration and hypothesis testing. In a typical data slicing workflow, experts run a slice discovery algorithm once and then inspect only the output slices. Prior studies [JCPT23, BRYB23] emphasize the need for visual analytics systems that effectively present slices to users, while accounting for cognitive load and enabling efficient slice navigation. Moreover, since automated methods may not cover all scenarios of interest [GILR23], interactive workflows could be designed to leverage the stakeholder's domain knowledge in order to define coherent subsets of data [JCPT23].

4.2. Design Requirements

R1. Automate slice discovery for object detectors beyond metadata and visual concepts. Given the limitations of prior methods (**C1**), our framework should extract more discriminative features in order to slice the data. Each slice should exhibit coherent error patterns, and metrics should be provided to assess its significance.

R2. Provide natural language explanations for slices. Since slice interpretation is labor-intensive and prone to errors (**C2**), our framework should generate descriptive sentences that explain slice scenarios and root causes, aiding experts in their reasoning process. We choose free-text explanations as they can effectively describe nuanced and complex image features and are inherently interpretable by humans [MBP*20, MBDP22].

R3. Support efficient slice inspection and refinement. Considering the substantial efforts involved in reviewing data slices (**C3**), our framework should effectively represent them to users. The system should (**R3-1**) offer concise overviews of data slices, (**R3-2**) highlight key statistics to help users identify slices of interest, (**R3-3**) provide detailed slice information on demand, and (**R3-4**) allow users to edit slices and explanations in case of inaccuracy.

R4. Enable interactive validation of user-defined slices. Beyond existing slices, our framework should enable users to test hypothesis (**C3**) by (**R4-1**) discovering new slices via the visual interface and (**R4-2**) defining slices with natural language descriptions. Once a hypothetical slice is created, our framework should compute slice metrics and explanations to facilitate hypothesis validation.

5. The *VISLIX* Framework

Guided by the design requirements, we develop *VISLIX*, a framework that automatically identifies data slices in object detectors, explains them in natural language, and enables expert exploration and hypothesis testing via a visual analytic system. Our framework uses large foundation models, such as VLMs for bridging images and text, and LLMs for reasoning about slice errors. The *VISLIX* workflow (Fig. 2) processes TPs, FPs, and FNs from a trained object detector (e.g., a car detector) on validation images in three phases:

Slice Finding. Each detection is converted into a context-aware embedding, followed by dimensionality reduction and clustering of FP and FN embeddings to identify slices. Slice importance is then estimated using nearby TPs (Sec. 5.1, addressing **R1**).

Slice Explanation. Assisted by VLMs and LLMs, we generate a free-text explanation for each slice by first producing textual explanations for each FP and FN within the slice, and then summarizing them into a slice explanation (Sec. 5.2, addressing **R2**).

Slice Exploration. We build a visual system that integrates slices and their explanations, enabling users to explore slices, rectify errors, test hypotheses, and enhance model performance (Sec. 5.3, addressing **R3** and **R4**). At the project's inception, we utilized VLMs like BLIP-2 [LLSH23] and LLaVA [LWZ*24] and LLMs

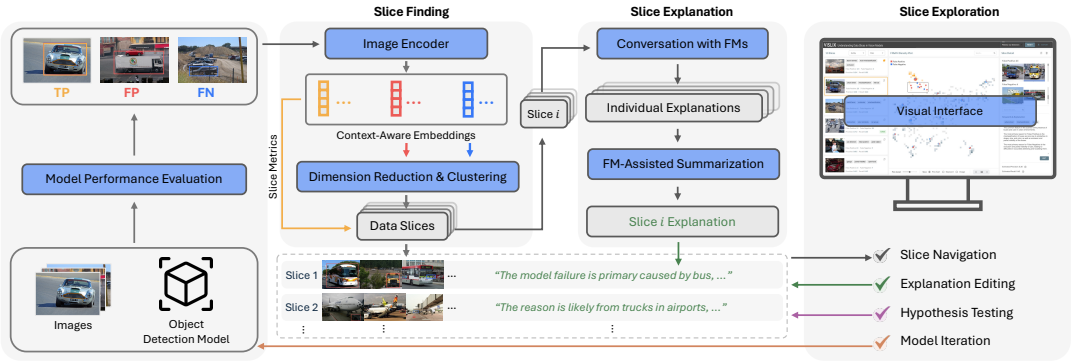

Figure 2: VISLIX *workflow.* ***Inputs****: True Positives (TPs), False Positives (FPs), and False Negatives (FNs) from an object (car) detector and validation images.* ***Slice finding****: Identifying data slice via image embeddings of FPs and FNs and estimating slice metrics via TPs.* ***Slice explanation****: Producing free-text explanations for each slice based on individual explanations of FPs and FNs, leveraging foundational models.* ***Slice exploration****: A visualization system that integrates all slices and explanations, enabling slice examination and validation.*

like GPT 3.5 [Ope23]) for their competitive performances, but *VISLIX* can easily integrate new models to keep up with model advancements.

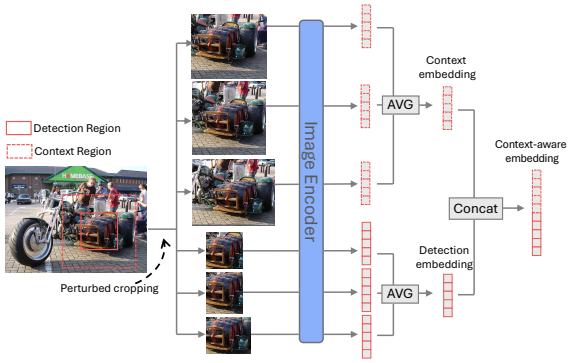

Figure 3: *Context-aware embedding generation.*

5.1. Slice Finding

To identify data slices, we combine the cropping strategy [LOW*20] that is widely used in computer vision to extract objects' local context, and embedding-based slice-finding methods [EVS*22, ddWLB22] that excel in image classifiers.

Context-aware embeddings. For each FP, FN, and TP, we construct a *context-aware* embedding by fusing features of the detection and its surroundings. This is achieved by enlarging the detection window and concatenating embeddings, a common practice in object detection [GK15, LWL*16, ZZW*17]. As illustrated in Fig. 3, for an FP detection featuring three barrels, we create a context region twice the size of the detection window, as per Zhu et al. [ZZW*17]. The enlarged context reveals that the barrels are in a motorcycle sidecar, likely causing the error. To enrich feature diversity, we crop three patches for each region with random perturbations of up to 10% expansion per side. The patch embeddings, obtained via BLIP-2's image encoder, are averaged to create the *detection* or *context* embeddings. These are concatenated to generate the final *context-aware* embedding, highlighting the barrels within the broader sidecar context to aid error reasoning.

Clustering-based slice discovery. We identify data slices as dense regions in the *context-aware* embedding space of FPs and FNs, which reveal shared features indicative of systematic errors. Specifically, we reduce embeddings to 10 dimensions using UMAP [MHM18], following prior work [MSRPC21, SDA*20] that demonstrates UMAP's effectiveness in cluster detection. Next, we apply HDBSCAN [CMS13] to identify high-density clusters (i.e., non-noise instances) while excluding isolated failures (i.e., noise instances). Unlike DBSCAN, HDBSCAN can find regions with varying densities and requires fewer parameters. We tune the clustering parameters using Silhouette [Rou87], informed by prior studies [MSRPC21, SDA*20]. Details are offered in the supplement. Fig. 9 illustrates two slices from a car detector: one in an airport setting and another featuring motorcycles.

Slice metrics. For every data slice, we compute two performance metrics: precision and recall. To identify the TP samples belonging to the slice, we measure the Euclidean distance between the slice and its nearby TPs in the *context-aware* embedding space. For each slice, we compute the average nearest neighbor distance of each instance within that slice. A TP is assigned to the slice if its distance to the nearest instance in the slice is smaller than this average nearest neighbor distance. We then calculate the precision and recall to assess the significance of each slice, giving more attention to those with lower values.

5.2. Slice Explanation

The slice explanation is generated in two steps: first, individual explanations for FPs and FNs are computed using a VLM (LLaVA) and an LLM (GPT 3.5, hereafter GPT). Then, the individual explanations are summarized into a cohesive slice explanation.

Individual explanations. To generate a natural-language explanation for an FP or FN, we first gather textual information about the image that may have caused the error. As shown in Fig. 4, we crop multiple patches to improve analysis accuracy: the detection region (DR), representing

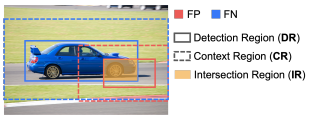

Figure 4: *Region annotations.*

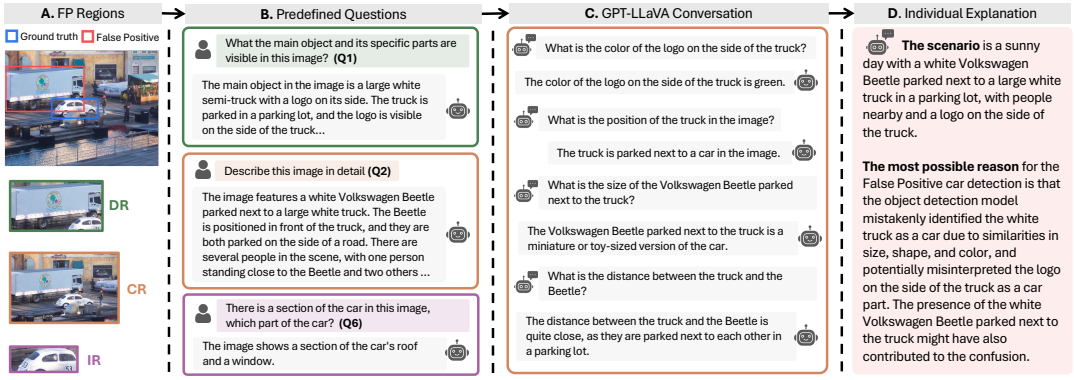

Figure 5: *Explanation generation for an FP in a car detector. **A**: Three regions derived from the FP: detection region (DR), context region (CR), and intersection region (IR). **B**: LLaVA answers predefined questions regarding different regions. **C**: GPT uses answers from **B** to chat with LLaVA about the CR. **D**: GPT explains the FP based on all the acquired information.*

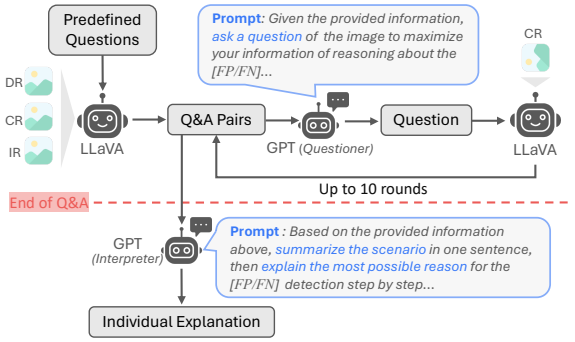

Figure 6: *The process of generating an individual explanation.*

Regions	Predefined questions
DR	**Q1.** FP: What the main object and its specific parts are visible in this image? FN: Describe the [*obj*] shown in the image and evaluate whether the entire [*obj*] is clearly visible.
CR	**Q2.** Describe this image in detail. **Q3.** What is the weather in this image? **Q4.** How is the lighting condition in this image? **Q5*.** Is the view of the [*obj*] in the image obstructed? If so, what is obstructing it?
IR	**Q6*.** There is a section of the [*obj*] in this image, which part of the [*obj*]?

Table 1: *Predefined questions for detection region (DR), context region (CR), and intersection region (IR). Q5 and Q6 are used exclusively for instances with IoU* > 0.2.

the predicted region for an FP or the ground truth region for an FN; the context region (CR), which expands the detection region by $2\times$ following [ZZW*17] (see Sec. 5.1); and the intersection region (IR), the overlap between the detection and ground truth when IoU exceeds 0.2 based on our preliminary experiments, indicating partial detection. An example is shown in Fig. 5-A.

Fig. 6 shows the procedure for generating an individual explanation. We first acquire answers from LLaVA to several predefined questions (Tab. 1) tailored for different regions that are generally critical for error reasoning. For instance, we prompt LLaVA to identify objects in the detection region of an FP (Q1) that may have caused the misclassification, describe the context region (Q2) for scene understanding, or describe the detected portion of the target object in the intersection region (Q6) to analyze partial detection. Fig. 5-B illustrates LLaVA's responses to these queries.

Since various factors can influence detection accuracy, a fixed set of universal questions is impractical. Building upon prior work [ZCH*23] that demonstrates GPT's ability to generate insightful questions with appropriate prompts, we prompt GPT (*Questioner*) with question-answer pairs to formulate questions aimed at uncovering the error's root cause. This question is then posed to LLaVA, and its response is fed back to GPT for further question generation. This iterative dialogue continues until GPT outputs "STOP" or completes 10 iterations, following [ZCH*23] (Fig. 5-C); Finally, GPT (*Interpreter*) summarizes the interactions between LLaVA, predefined questions, and GPT into a concise explanation of the scene and its primary cause. Fig. 5-D illustrates an FP explanation. All explanations of FPs and FNs are precomputed.

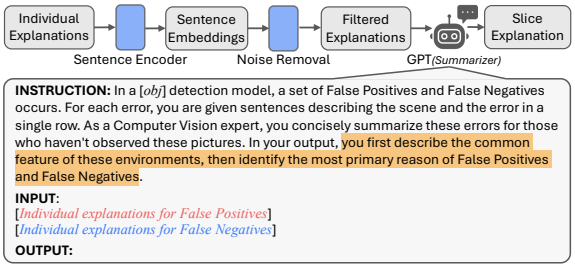

Figure 7: *Pipeline for generating a slice explanation from individual explanations and the prompt template used for GPT.*

Summarizing individual explanations. We derive slice explanations by aggregating individual FP and FN explanations using GPT (*Summarizer*). However, these explanations can be inconsistent or erroneous due to hallucinations or outliers, affecting quality. To address this, as shown in Fig. 7, we convert explanations into text embeddings with a sentence transformer, then select those closest to the centroid, ensuring up to 80% of the slice is represented or the total token count stays within 2000 (input length limit). Using these filtered explanations, GPT (*Summarizer*) generates a description of the scene and identifies root causes for FPs and FNs. Fig. 9 shows two slices from a car detector with their explanations. Full prompts and additional examples are provided in the supplement.

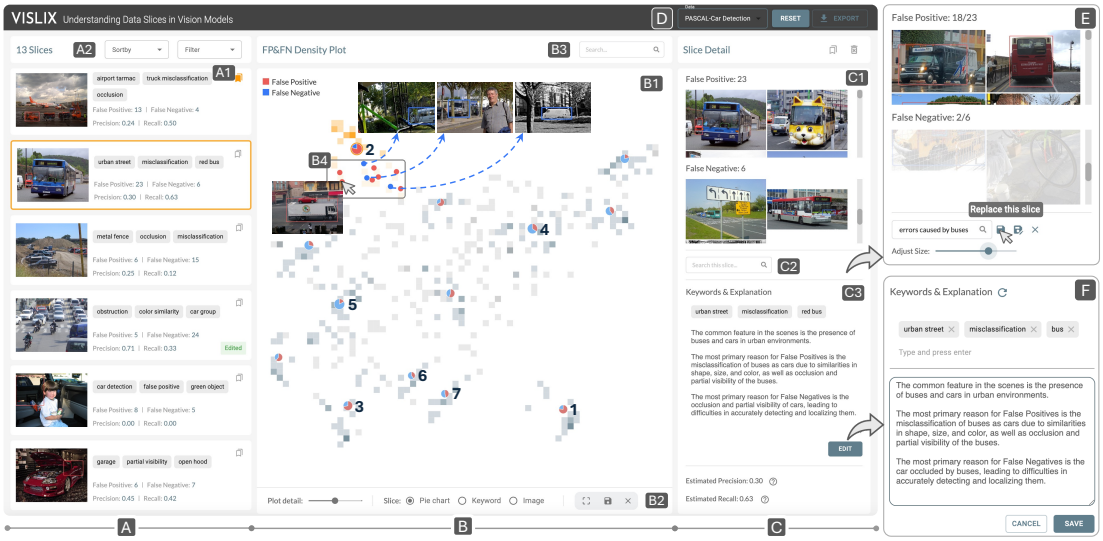

Figure 8: *The* VISLIX *interface visualizes slices and explanations from a car detector.* ***Slice list view (A)*** *shows all slices as summary cards (A1), with sorting and filtering options (A2).* ***Slice plot view (B)*** *includes a density plot (B1) showing slice locations in the embedding space (with slices 1–7 annotated for reference), an editing bar (B2) for operations like brushing to create hypothetical slices (B4), and a global search bar (B3) for hypothetical slice creation via queries.* ***Slice detail view (C)*** *presents details of a selected slice (slice 2), supporting query-based refinement (E) and explanation editing (F).* ***Toolbar (D)*** *enables data switching, operation resets, and slice export.*

Figure 9: *A random sample of FPs and FNs in slices 1 (top) and 3 (bottom) and their explanations in the car detector.*

5.3. Slice Exploration Through an Interactive System

To enable users to analyze the generated slices and explanations (**R3**) and test slice hypotheses (**R4**), we present a visual analytics system (Fig. 8) with four linked views: slice list view (A), slice plot view (B), slice detail view (C), and a toolbar (D). View A lists all produced slices (**R3-1**), summarizing key information in cards (Fig. 8-A1). To prevent overwhelming users, each card shows three keywords that capture the slice's scenarios and error causes, generated by GPT using the same input as the slice explanation (Fig. 7) but with modified task descriptions. Further details are provided in the supplement. In the following sections, we detail views B and C and discuss our key design decisions in between.

5.3.1. Slice Plot View

The **2D density plot** (Fig. 8-B1) provides a slice overview in the embedding space using the 2D dimensionality reduction result of UMAP applied to the *context-aware* embeddings of FPs and FNs. To better reflect the data distribution used in slice discovery (Sec. 5.1), UMAP parameters are largely preserved, with the component number reduced from 10 to 2 and the minimum distance increased to 0.15 to mitigate visual clutter.

Next, we divide the scatterplot into a 64×64 grid by default and calculate the densities of noise and non-noise instances, as determined by HDBSCAN (Sec. 5.1), in each cell. These densities are then mapped to two distinct colormaps—gray for noise and blue (or orange, its complementary color, when the slice is selected) for non-noise—and blended together as the grid rendering. Thus, this plot characterizes the data distribution while emphasizing slices. We employ the density plot as the primary visualization due to its scalability for dense scatterplots and, more importantly, its intuitive representation of HDBSCAN's mechanism, where clusters emerge from high-density areas, thus enhancing user trust in the data slices. For instance, variations in cluster area and color indicate slice size and compactness, whereas spatial locations reflect slice similarities, enabling users to quickly pinpoint slices of interest and facilitate a straightforward comparison among slices (**R3-1** and **R3-2**). We discuss alternative design in the supplement.

Editing bar (Fig. 8-B2) offers three operations on the density plot. On the left, users can adjust the granularity (grid resolution) of B1 to adapt to varied data scales. In the middle, three types of slice encodings are offered based on experts' suggestion: pie charts (depicting slice size and the percentage of FPs and FNs), the first slice keyword, and one representative image. Each encoding is centered within its respective slice. Further, the slice list view (Fig. 8-A) and plot view (Fig. 8-B) are cross-filtered: selecting a slice in view B highlights both its density area and the corresponding slice card in view A in orange, and vice versa (**R3-2**).

With the brush tool (Fig. 8-B4), users can select an area of in-

terest by clicking and dragging, which creates a semitransparent overlay on the view. FP and FN instances within the selected region are displayed as red and blue circles, respectively. Hovering over a point reveals its corresponding image. If users identify consistent patterns in this area, they can click the save icon to generate a hypothetical slice. The backend computes the slice's statistics and explanations, which are then displayed in the interface. Users can choose to keep or delete it once they test their hypotheses (**R4-1**).

Global search bar (Fig. 8-B3) allows experts to create hypothetical slices via textual queries (**R4-2**). As shown in Fig. 11, after users input a scenario description, the system retrieves instances matching the query based on cosine similarity between sentence embeddings. Embeddings of individual explanations are precomputed and stored in the backend. By default, instances with a similarity score above 0.5 are returned. A slider lets experts adjust the similarity threshold to refine results. Filtered FPs and FNs appear as circles on the density view, with images revealed on mouseover. If the results align with expectations, users can save the slice.

5.3.2. Slice Detail View

The slice detail view (Fig. 8-C) provides details about the selected slice (**R3-3**). C1 displays FPs and FNs with bounding boxes rendered on full images, and allows users to toggle between the full image and a magnified detection area. C2 includes a search box for refining slices by querying specific scenarios; filtered instances are determined by query similarity, and highlighted in the density plot and view E. Users can save these results as a new slice or replace the current one. C3 shows keywords, explanations, and slice metrics. To address potential hallucinations, users can switch to editing mode (Fig. 8-F) by clicking "Edit," where they can refresh keywords and explanations via GPT or manually edit them (**R3-4**).

6. Expert Study

To evaluate *VISLIX*, we conducted a series of structured interviews with six ML experts (E1 - E6), who were recruited by e-mail. They are not coauthors of this paper and have not previously seen *VISLIX*. The panel consisted of industry ML practitioners (researchers and engineers), with an average age of 33.167 ± 5.707 years, and varying STEM education backgrounds: four holding doctoral degrees, one master's degree, and one bachelor's degree in Computer Science. All experts have worked in the field of Advanced Driver Assistance and Autonomous Driving Systems (ADAS/AD), and have had prior experience in developing object detection models. Collectively, these six experts possess an average of 6.667 ± 3.077 years of experience in the field.

Procedure. Experts were asked to analyze edge cases for four object detection models: "car," "person," "chair," and "dog". We divided the experts into two groups to evaluate both *VISLIX*'s effectiveness and, exclusively, the explanation quality. The first group used *VISLIX* to analyze "car" and "person" detectors and rated textual explanations for "chair" and "dog" on a 5-point Likert scale, while the second group did the reverse. During the practical part of the study, experts received a ten-minute tutorial, explored the system for thirty minutes using the "think-aloud" protocol, and completed a post-study questionnaire.

This section is organized as follows: we first describe the model and data used in the interviews. Then, we describe three use cases derived from the expert study, where our system was used to inspect the detectors. Finally, we summarize the experts' feedback.

6.1. Model and Data Description

We use the PASCAL image dataset [EVGW*] and select four objects from differing categories with varying sizes: person (Person), dog (Animal), car (Vehicle), and chair (Indoor). For each object, we first extract images containing it, and then split them into training and validation sets with a ratio of 60% to 40%, respectively. We train a detector for each object using a popular object detection model, Fast R-CNN [Gir15], and end the training process when the loss falls below 0.05 or after 40 epochs. Then, we obtain detections (TPs, FPs, and FNs) of the validation set for each detector, and execute *VISLIX* once to generate data slices, slice explanations, and integrate them into the visual analytics system. We report the statistics for each detector in Tab. 2.

Object	Ground truth	mAP	TPs	FPs	FNs	# Slices
Dog	1227	86.71%	541	165	64	7
Car	2364	75.56%	752	186	209	13
Chair	2906	47.83%	668	481	530	15
Person	10129	75.42%	3217	2066	808	24

Table 2: *Detector statistics: ground truth size, mAP, sizes of TPs, FPs, FNs on the validation set, and the number of output slices.*

6.2. Use Case 1: Slice Analysis of a Car Detector

E1, E3, and E5 would like to identify systematic errors in a car detector in order to write a validation report. *VISLIX* identified 13 data slices (Tab. 2), some of which are shown in Fig. 8.

Slice understanding. When examining slice 1, all experts recognize the FPs and FNs mainly occur at the airport. E3 notices that this slice is distant from others in the density plot view. The textual explanation (Fig. 9) further states that FPs are caused by misclassification of "trucks" or "airplanes" in airports as cars, whereas FNs are caused by trucks or luggage carts obstructing cars. The experts agree with the explanation but provide additional causes. For example, E3 identified some labeling issues: *"The model is identifying a car, but the labeler did not."*

Slice refinement. E5 investigates slice 2 highlighted in Fig. 8, and understands the primary cause is misclassification or occlusion caused by buses. E5 observes several FNs incorrectly labeling trucks as cars (Fig. 8-B4). E5 believes the mislabeling issue in the data needs to be fixed and thus saves several truck images. To refine this slice, the user searches for *"errors caused by buses"* (Fig. 8-E), adjusts the range bar to exclude trucks and clicks the "Replace" button to replace the original slice. After clicking the "Edit" button, the user may change the keyword "red bus" to "bus" and modify explanation slightly for a more accurate description (Fig. 8-F).

Slice navigation. E1 sorts slices (Fig. 8-A1) by precision and notices that slice 3 has a low precision (0.12), indicating FPs are likely to occur. The slice examples and explanations are shown in Fig. 9, suggests that the detector often misclassified motorcycles as cars, or that motorcycles obstructed the car.

Through iterative analysis, E1 saves more slices of interest (Fig. 10). For example, slice 5, with the largest number of FNs, mainly occurs on streets with overlapping cars. The error is likely caused by low labeling quality, as described by the explanation, *"the ground truth car is partially obstructed by other objects such as buses, cars, buildings, fences, mirrors, windows."* Slices 6 and 7, spatially close in the plot view, mostly appear in indoor and outdoor commercial scenes such as exhibitions, typically with open car hoods. These slices present unusual car states, and contain out-of-distribution examples where the detector fails.

Figure 10: *Car detector: FPs and FNs in slices 5, 6, and 7.*

Hypothesis testing. While investigating the data slices, E5 finds some images containing snow, and wants to investigate whether snow negatively impacted the car detection. To test their hypothesis, as shown in Fig. 11, E5 globally queries *"Car in the snow weather"* and then adjusts the slider to include 5 FPs and 8 FNs to form a hypothetical slice that has a precision of 0.38 and a recall of 0.27. From the instances and explanations, E5 deduces that cars covered by snow or tree branches, in low visibility, can impact the model performance.

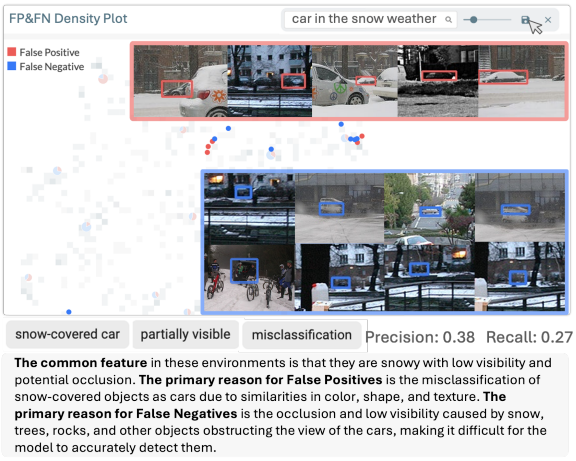

Figure 11: *Hypothetical slice in snowy weather created by queries.*

6.3. Use Case 2: Improving a Person Detection Model

The same experts, E1, E3 and E5, also use *VISLIX* to analyze a person detection model, as shown in Fig. 12. After adjusting the plot level of detail for a more granular inspection of the embedding (Fig. 12-A3), E3 notices a dense cluster on the left side of the view that the algorithm does not identify as a slice. Using the brushing tool, E3 observes numerous images featuring horses and persons (Fig. 12-A1) and posits it as a missing slice. Thus, he creates a new slice for these predictions to validate his hypothesis (Fig. 12-A2). This new slice, shown in Fig. 12-B, has a low precision of 0.32, and the explanation suggests the failure reason to be misclassification or occlusion caused by horse features such as heads and legs. This slice is then saved for further analysis. E1, E3, and E5 also identify other critical slices based on their experience, and all saved slices (1-8) are shown in Fig. 12. These slices typically have coherent explanations and are caused by occlusion or inaccurate detections in diverse scenarios such as bicycle (1) and boat (2) scenes, people inside vehicles (3, 4), infants (6), and indoor scenarios (5, 7).

Based on the feedback from the users, we fine-tune the model with a focus on these eight slices. The strategy is to retrieve images from training data that resemble the images in these slices, and then use them to train the model with one more epoch. Specifically, we employ two approaches to extract images. The first is embedding-based, where we export the slice images from *VISLIX* and convert them into embeddings through the encoder of BLIP-2. Then, we compute the embedding center and select the training images whose embeddings are closest to the center. The second approach uses CLIP scores [RKH*21], which measure the alignment between images and text descriptions and have been widely used in image retrieval [JCPT23]. Following Johnson et al. [JCPT23], we use the template *"A photo of [...]"*, together with the slice description generated by *VISLIX*. Then, we extract training images that have higher CLIP scores with the slice description. For each approach, we curate a new training set.

The model is then trained for one more epoch using the re-weighting technique [IAPLP22] for worst-group optimization: we select three times the number of images for each slice, combine the original eight slices, and replicate them to match the size of the original training set. The retraining is done on three datasets: the original dataset, the embedding-based augmented dataset, and CLIP score-based training dataset. We refer to three conditions as *Original*, *Embedding*, and *CLIP Score*. We then validate each model and recalculate the slice metrics (precision and recall) and model metrics (precision, recall, and mAP). Slice metrics are approximated by considering all objects in these images. Overall, across all three conditions, the *Embedding* and *CLIP Score* conditions demonstrate the best trade-off between slice precision and recall, achieving comparable mAP values of 0.7878 and 0.7845, respectively—approximately 3% higher than the baseline mAP of 0.7542 before fine-tuning. In contrast, the *Original* condition yields a mAP of 0.7265, falling below the baseline. Notably, the *CLIP Score* condition exhibits the best slice-level performance, with 5 slices showing the biggest improvements in precision and another 5 slices showing the biggest improvements in the recall. This might be due to the fact that *CLIP Score* retrieves more diverse images than *Embedding*. We offer the detailed results in the supplement.

6.4. Use Case 3: Analysis of a Dog and Chair Detector

E2, E4, and E6 analyze the detectors for the classes "dog" and "chair". Here, we summarize their insights regarding the model's edge cases. **Dog detector**: All experts quickly observe that the model frequently confuses dogs with other animals. E2 notes, *"The explanations are really good: sheep and cows recognized as a dog."* Additionally, E4 and E6 identify a recurring issue when a person is

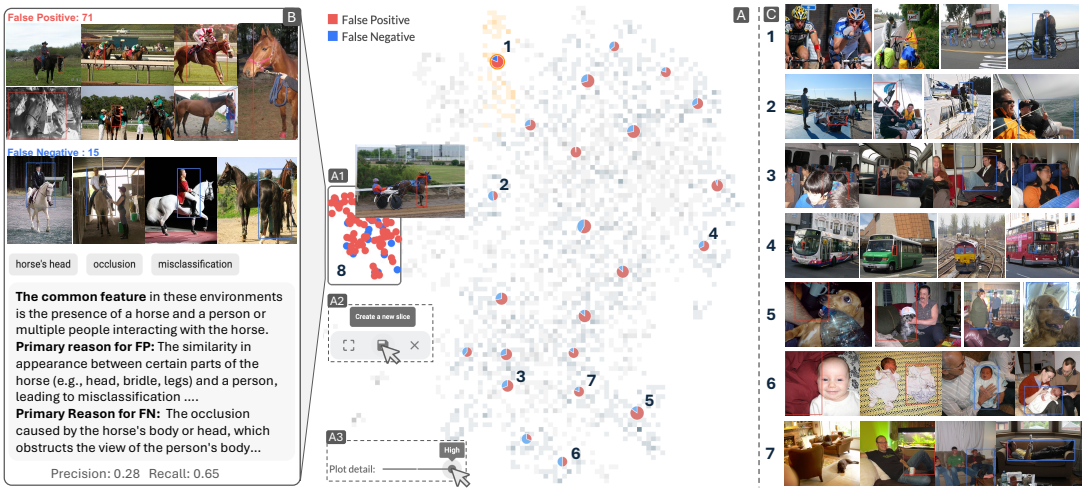

Figure 12: *Illustration of the slices in the person detector used in use case 2.* ***(A)*** *The density plot view with a high plot detail (A3). The area A1 is brushed to form a new slice (A2).* ***(B)*** *The new slice information, such as a sample of False Positives (FPs) and False Negatives (FNs), and explanations.* ***(C)*** *Shows two FPs and two FNs for each of slices 1-7. The corresponding slice numbers (1-8) are annotated in (A).*

playing with a dog, where occlusion by the person often causes bounding box detection errors. **Chair detector**: All experts agree that occlusion by people sitting on chairs is the primary cause of most model failures. As E4 explains, *"An obvious thing would be people sitting on chairs."* However, other factors also affect detection performance. A noteworthy observation is that chairs with unique designs are often missed. E2 remarks, *"The false negative cases are chairs that look different and have some different kind of design."* Furthermore, E6 identifies couches as a frequent source of mistakes. After querying for *"couches,"* he suggests that many false positives could be attributed to inconsistent labeling.

6.5. Expert Feedback

Experts have praised *VISLIX*, noting its user-friendliness and functionality. E1 highlights the system's ability to provide insightful findings from edge cases, enhancing hypothesis testing: *"The initial findings from various edge cases provide valuable insights into failure scenarios. Delving into these groups has inspired me to uncover further reasons, enabling easy testing of my hypotheses."* Furthermore, E2 and E5 appreciate the quick access to detailed and organized data slices, which facilitates a deeper understanding of the model. E2 mentions that *"the slices are clustered nicely and are visually and semantically consistent"*. The system's intuitive user interface is praised by E3: *"It is simple to use and pretty easy to adapt to."* E4 and E6 remark on the flexibility and capabilities of the tool for exploring the model's mistakes. E6, in particular, highlights the system is a *"very intuitive tool to understand the edge cases and hone in on them based on the explanations provided"*. Overall, *VISLIX* has been recognized for its effective design and functionality, aiding users in efficiently navigating and analyzing complex data scenarios.

VISLIX has generally been well-received, but it has also drawn some constructive criticism from experts. E1 and E5 observe that the explanations provided by the system can be superficial and may fail to address the specific reasons behind the edge cases. E2 and E4 raise concerns about limitations due to dataset quality, such as inconsistent labeling. However, they also note that the *VISLIX* could be used to identify these inconsistencies and improve data label quality. E5 and E6 suggest enhancements to the user interface, particularly for image navigation, such as zooming and panning, and displaying ground truth and predictions separately to better support detailed examination of the data. These insights shed light on areas for potential refinement to enhance the system's capabilities and user experience. Experts have also evaluated the *VISLIX* tool and explanations using a 5-point Likert scale. The results reflect a positive overall impression of *VISLIX*; see the supplement for details.

Figure 13: *Examples of two slices with low-quality FP explanations. (a) a dog detector. (b) a chair detector.*

Explanation evaluation. We further analyze explanations with strong disagreement or average scores below neutral. One notable issue is model hallucinations: as shown in Fig. 13, the explanation of slice (a - dog detector) states, *"model mistaking objects or elements in the scene, such as towels, pillows, patterns, or furniture, as part of the dog"*, whereas partial detection and occlusions of towels or pillows are more likely the reason. However, we find most individual explanations are sound, frequently mentioning *"only part of the dog is detected"* and surroundings of *"towel, couch, and pillow"*. We speculate that LLM incorrectly connects these elements. Additionally, we notice that *VISLIX* struggles to interpret small detection: Fig. 13 slice (b - the chair detector) is wrongly explained as *"the model mistaking objects or textures in the scene, such as fabric, spheres, reflective surfaces, or wigs, as chairs"*. We attempted to refine the prompt by including the ratio of the detection region to the image, expecting the explanation would mention size-related challenges. However, we found that the model is often insensitive to such a ratio, misinterpreting a tiny detection as a large portion of the image. An alternative is to explicitly show the detection size in the system to inform experts, which we leave for future work.

6.6. Comparison with Other Slice Discovery Systems

The most relevant work to ours is *ConceptSlicer* [ZPOH*24], which identifies slices in object detectors using visual concepts. We run *ConceptSlicer* on the car detector (Sec. 6.2) and identify 9 slices with accuracy at least 5% below average. We then remove visually inconsistent slices and merge highly overlapping ones caused by concept quality issues (e.g., a single label for different object types or multiple labels for the same object). Fig. 14 shows samples from the resulting three slices, which align with the three slices identified by *VISLIX*: airplanes (Fig. 9-1), buses (Fig. 8-2), and motorcycles (Fig. 9-3), respectively.

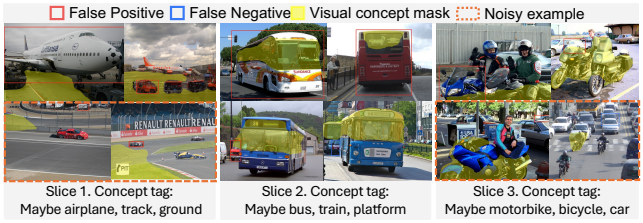

Figure 14: *Samples (FPs and FNs) of three slices in the car detector identified by* ConceptSlicer. *The relevant visual segments are masked in yellow. Spurious segments that are irrelevant to the detection are highlighted with dotted boxes.*

We observe that *VISLIX* often produces finer-grained slices. For example, *ConceptSlicer* groups all motorbike-present images regardless of detection relevance (Fig. 14-3), while *VISLIX* isolates motorbike misclassification or occlusion cases, e.g., motorbike/person interactions (Fig. 9-3). Similarly, *ConceptSlicer* merges distinct but visually similar scenarios like "airport runways" and "race tracks" (Fig. 14-1), whereas *VISLIX* separates them (Fig. 9-1). Moreover, *VISLIX* identifies slices missed by *ConceptSlicer* due to its reliance on visual concepts. For instance, slice 5 (Fig. 10, "car overlap") and slices 6 and 7 (missing concepts like "exhibition" and "car hood") are absent in *ConceptSlicer*. Abstract factors like image quality and weather are also overlooked by *ConceptSlicer*, missing slices for small detections and snowy conditions (Fig. 11). These examples show that visual concepts alone struggle to capture key features like image context (e.g., location, environment) and object relationships (e.g., proximity, positions). *VISLIX* bridges this gap by leveraging foundation models to explain detection mistakes.

We also compare *VISLIX* with *AttributionScanner* [XOG*25], a metadata-free method using attribution-weighted embeddings to detect slices in image classifiers. Adapting *VISLIX* to the landbird/waterbird classification task, we found 7 problematic slices for *AttributionScanner* and 5 for *VISLIX*. Our results suggest *AttributionScanner* focuses on spurious background correlations, while *VISLIX* captures and explains nuanced patterns, revealing model vulnerabilities to bird species, scenes, spurious correlations, and labeling issues. Further details are provided in the supplement.

7. Discussion and Future Work

Improving the slice discovery method. In *VISLIX*, the context region is obtained by enlarging the detection window twice. Although this strategy aligns with the previous work [ZZW*17], it may still capture insufficient context, particularly for small detections, resulting in inconsistent slices. A cropping strategy that prioritizes detection while maintaining sufficient context is needed. Furthermore, enabling user control of UMAP and HDBSCAN parameters could be a future enhancement [JCPT23], but it must carefully balance added complexity and computational cost.

Enhancing the quality of slice explanations. We break down the slice explanation task into three steps (Sec. 5.2): individual explanations, noise reduction, and explanation aggregation. We leverage prompt engineering [CZLZ23] (e.g., chain-of-thought reasoning, role-playing) and integrate knowledge of common detection errors. While these improve explanation quality, model hallucinations may still cause ungrounded explanations, such as confusing occlusion with misclassification or misinterpreting tiny detections. To address this, future work includes user-modifiable prompts, golden explanations as few-shot examples [DLD*22], and TPs as contrastive examples [RMP21]. Additionally, predefined questions (Tab. 1) may not generalize; e.g., the "weather" question (Q3) is unsuitable for indoor settings. Allowing experts to tailor core questions to applications is another way to improve explanations.

Improving the visual analytics system. The *VISLIX* interface helps users extract insights that are otherwise difficult to obtain. For example, as shown in Sec. 6.2, the density plot (Fig. 8) allows experts to identify isolated slices (slice 1), detect noise (2), and recognize visually similar slices (6, 7) by analyzing cluster density and distribution, enhancing slice inspection. Hypothesis testing via query (Fig. 11) and visual selection (Fig. 12) further aids model validation by uncovering previously unnoticed slices. Inspired by expert feedback and recent research, we plan to enhance our system by (1) displaying ground truth for incorrect detections, enabling the identification of mislabeled data, and (2) expanding the search space beyond the validation set (e.g., via image retrieval) to improve hypothesis testing and enable model refinements with additional training data [BHM23, WAG22, ZPOH*24].

Extending to other vision tasks. We plan to generalize *VISLIX* to tasks like semantic segmentation by using segment contours for the future. However, defining context and intersection regions for irregularly shaped segments requires careful consideration.

8. Conclusion

We present *VISLIX*, an XAI framework that leverages VLMs and LLMs to validate computer vision models in a human-in-the-loop manner. *VISLIX* automatically discovers and explains data slices, particularly in object detectors, and empowers experts to efficiently explore slices and interactively test hypothetical slices through a visual analytics system. Additionally, this framework is model-agnostic and can continually benefit from the ongoing advancement of foundation models.

Acknowledgments. This work started when X. Yan, X. Xuan, and L. Gou worked with Bosch Research North America. It was partially supported by NSF grants IIS-2205418 and DMS-2134223, and a seed grant from the Utah Board of Higher Education's Deep Technology Initiative.

References

[AGM*18] ADEBAYO J., GILMER J., MUELLY M., GOODFELLOW I., HARDT M., KIM B.: Sanity checks for saliency maps. *Advances in neural information processing systems 31* (2018). 2

[ALXD23] AHN Y., LIN Y.-R., XU P., DAI Z.: 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* (2023), pp. 1–16. 2

[AS*94] AGRAWAL R., SRIKANT R., ET AL.: Fast algorithms for mining association rules. In *Proceedings of 20th International Conference on Very Large Data Bases (VLDB)* (1994), vol. 1215, Santiago, Chile, pp. 487–499. 2

[BGM*17] BAILIS P., GAN E., MADDEN S., NARAYANAN D., RONG K., SURI S.: Macrobase: Prioritizing attention in fast data. In *Proceedings of the 2017 ACM International Conference on Management of Data* (2017), pp. 541–556. 2

[BHM23] BOREIKO V., HEIN M., METZEN J. H.: Identifying systematic errors in object detectors with the scrod pipeline. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 4090–4099. 1, 10

[BRYB23] BALAYN A., RIKALO N., YANG J., BOZZON A.: Faulty or ready? handling failures in deep-learning computer vision models until deployment: A study of practices, challenges, and needs. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (2023), pp. 1–20. 1, 2, 3

[BSI*24] BORDES F., SHEKHAR S., IBRAHIM M., BOUCHACOURT D., VINCENT P., MORCOS A.: PUG: Photorealistic and semantically controllable synthetic data for representation learning. *Advances in Neural Information Processing Systems 36* (2024). 1

[CBL*20] CAESAR H., BANKITI V., LANG A. H., VORA S., LIONG V. E., XU Q., KRISHNAN A., PAN Y., BALDAN G., BEIJBOM O.: nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2020), pp. 11621–11631. 1

[CDHP21] CABRERA Á. A., DRUCK A. J., HONG J. I., PERER A.: Discovering and validating AI errors with crowdsourced failure reports. *Proceedings of the ACM on Human-Computer Interaction 5*, CSCW2 (2021), 1–22. 2, 3

[CFB*23] CABRERA Á. A., FU E., BERTUCCI D., HOLSTEIN K., TALWALKAR A., HONG J. I., PERER A.: Zeno: An interactive framework for behavioral evaluation of machine learning. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (2023), pp. 1–14. 2

[CGT*24] CHEN C., GUO Y., TIAN F., LIU S., YANG W., WANG Z., WU J., SU H., PFISTER H., LIU S.: A unified interactive model evaluation for classification, object detection, and instance segmentation in computer vision. *IEEE Transactions on Visualization and Computer Graphics 30*, 1 (2024), 76–86. 2

[CKP*19] CHUNG Y., KRASKA T., POLYZOTIS N., TAE K. H., WHANG S. E.: Slice Finder: Automated data slicing for model validation. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)* (2019), IEEE, pp. 1550–1553. 2, 3

[CMS13] CAMPELLO R. J., MOULAVI D., SANDER J.: Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining* (2013), Springer, pp. 160–172. 4

[CZLZ23] CHEN B., ZHANG Z., LANGRENÉ N., ZHU S.: Unleashing the potential of prompt engineering in large language models: a comprehensive review. *arXiv preprint arXiv:2310.14735* (2023). 10

[ddWLB22] D'EON G., D'EON J., WRIGHT J. R., LEYTON-BROWN K.: 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. 1, 2, 3, 4

[DLD*22] DONG Q., LI L., DAI D., ZHENG C., WU Z., CHANG B., SUN X., XU J., SUI Z.: A survey on in-context learning. *arXiv preprint arXiv:2301.00234* (2022). 10

[EVGW*] EVERINGHAM M., VAN GOOL L., WILLIAMS C. K. I., WINN J., ZISSERMAN A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 7

[EVS*22] EYUBOGLU S., VARMA M., SAAB K. K., DELBROUCK J.-B., LEE-MESSER C., DUNNMON J., ZOU J., RE C.: Domino: Discovering systematic errors with cross-modal embeddings. In *International Conference on Learning Representations* (2022). 1, 2, 3, 4

[GILR23] GAO I., ILHARCO G., LUNDBERG S., RIBEIRO M. T.: Adaptive testing of computer vision models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023), pp. 4003–4014. 2, 3

[Gir15] GIRSHICK R.: Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision* (2015), pp. 1440–1448. 7

[GK15] GIDARIS S., KOMODAKIS N.: Object detection via a multi-region and semantic segmentation-aware CNN model. In *Proceedings of the IEEE international conference on computer vision* (2015), pp. 1134–1142. 4

[GZL*20] GOU L., ZOU L., LI N., HOFMANN M., SHEKAR A. K., WENDT A., REN L.: VATLD: A visual analytics system to assess, understand and improve traffic light detection. *IEEE transactions on visualization and computer graphics 27*, 2 (2020), 261–271. 1

[HPY00] HAN J., PEI J., YIN Y.: Mining frequent patterns without candidate generation. *ACM sigmod record 29*, 2 (2000), 1–12. 2

[IAPLP22] IDRISSI B. Y., ARJOVSKY M., PEZESHKI M., LOPEZ-PAZ D.: Simple data balancing achieves competitive worst-group-accuracy. In *Conference on Causal Learning and Reasoning* (2022), PMLR, pp. 336–351. 8

[IBB*22] IDRISSI B. Y., BOUCHACOURT D., BALESTRIERO R., EVTIMOV I., HAZIRBAS C., BALLAS N., VINCENT P., DROZDZAL M., LOPEZ-PAZ D., IBRAHIM M.: ImageNet-X: Understanding model mistakes with factor of variation annotations. In *The Eleventh International Conference on Learning Representations* (2022). 3

[JCPT23] JOHNSON N., CABRERA Á. A., PLUMB G., TALWALKAR A.: Where does my model underperform? a human evaluation of slice discovery algorithms. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing* (2023), vol. 11, pp. 65–76. 1, 2, 3, 8, 10

[JLMM22] JAIN S., LAWRENCE H., MOITRA A., MADRY A.: Distilling model failures as directions in latent space. In *The Eleventh International Conference on Learning Representations* (2022). 2, 3

[KPSH21] KRISHNAKUMAR A., PRABHU V., SUDHAKAR S., HOFFMAN J.: UDIS: Unsupervised discovery of bias in deep visual recognition models. In *British Machine Vision Conference (BMVC)* (2021), vol. 1, p. 3. 2

[LLLL24] LIU H., LI C., LI Y., LEE Y. J.: Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 26296–26306. 2

[LLSH23] LI J., LI D., SAVARESE S., HOI S.: BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning* (2023), pp. 19730–19742. 2, 3

[LLWL24] LIU H., LI C., WU Q., LEE Y. J.: Visual instruction tuning. *Advances in neural information processing systems 36* (2024). 2

[LOW*20] LIU L., OUYANG W., WANG X., FIEGUTH P., CHEN J., LIU X., PIETIKÄINEN M.: Deep learning for generic object detection: A survey. *International journal of computer vision 128* (2020), 261–318. 2, 3, 4

[LWHC22] LEE S., WANG Z. J., HOFFMAN J., CHAU D. H. P.: VisCUIT: Visual auditor for bias in CNN image classifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 21475–21483. 2

[LWL*16] LI J., WEI Y., LIANG X., DONG J., XU T., FENG J., YAN S.: Attentive contexts for object detection. *IEEE Transactions on Multimedia 19*, 5 (2016), 944–954. 2, 4

[LWZ*24] LI C., WONG C., ZHANG S., USUYAMA N., LIU H., YANG J., NAUMANN T., POON H., GAO J.: LLaVA-Med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems 36* (2024). 3

[MBDP22] MARASOVIC A., BELTAGY I., DOWNEY D., PETERS M. E.: Few-shot self-rationalization with natural language prompts. In *2022 Findings of the Association for Computational Linguistics: NAACL 2022* (2022), Association for Computational Linguistics (ACL), pp. 410–424. 3

[MBP*20] MARASOVIĆ A., BHAGAVATULA C., PARK J. S., LE BRAS R., SMITH N. A., CHOI Y.: Natural language rationales with full-stack visual reasoning: From pixels to semantic frames to commonsense graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (2020), pp. 2810–2829. 3

[MHM18] MCINNES L., HEALY J., MELVILLE J.: Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018). 4

[MSRPC21] MCCONVILLE R., SANTOS-RODRIGUEZ R., PIECHOCKI R. J., CRADDOCK I.: N2D:(not too) deep clustering via clustering the local manifold of an autoencoded embedding. In *2020 25th international conference on pattern recognition (ICPR)* (2021), IEEE, pp. 5145–5152. 4

[Ope22] OPENAI: ChatGPT, 2022. Accessed: September 12, 2023. URL: https://chat.openai.com/. 2

[Ope23] OPENAI: GPT-4 technical report, 2023. arXiv:2303.08774. 2, 4

[PBdA*23] PASTOR E., BARALIS E., DE ALFARO L., 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.* (2023), IEEE. 2

[PDAB21] PASTOR E., DE ALFARO L., BARALIS E.: Looking for trouble: Analyzing classifier behavior via pattern divergence. In *Proceedings of the 2021 International Conference on Management of Data* (2021), pp. 1400–1412. 2

[RCO*19] RAHWAN I., CEBRIAN M., OBRADOVICH N., BONGARD J., BONNEFON J.-F., BREAZEAL C., CRANDALL J. W., CHRISTAKIS N. A., COUZIN I. D., JACKSON M. O., ET AL.: Machine behaviour. *Nature 568*, 7753 (2019), 477–486. 1

[RKH*21] RADFORD A., KIM J. W., HALLACY C., RAMESH A., GOH G., AGARWAL S., SASTRY G., ASKELL A., MISHKIN P., CLARK J., KRUEGER G., SUTSKEVER I.: Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning* (London, UK, 18–24 Jul 2021), Meila M., Zhang T., (Eds.), vol. 139 of *Proceedings of Machine Learning Research*, PMLR, pp. 8748–8763. 2, 8

[RMP21] ROSS A., MARASOVIĆ A., PETERS M. E.: Explaining nlp models via minimal contrastive editing (MiCE). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (2021), pp. 3840–3852. 10

[Rou87] ROUSSEEUW P. J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics 20* (1987), 53–65. 4

[SB21] SAGADEEVA S., BOEHM M.: 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. 1, 2

[SDA*20] SOHONI N., DUNNMON J., ANGUS G., GU A., RÉ C.: No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems 33* (2020), 19339–19352. 2, 4

[ŞSAAH*23] ŞENGÖNÜL E., SAMET R., ABU AL-HAIJA Q., ALQAHTANI A., ALTURKI B., ALSULAMI A. A.: An analysis of artificial intelligence techniques in surveillance video anomaly detection: A comprehensive survey. *Applied Sciences 13*, 8 (2023), 4956. 1

[TMS*23] TOUVRON H., MARTIN L., STONE K., ALBERT P., ALMAHAIRI A., BABAEI Y., BASHLYKOV N., BATRA S., BHARGAVA P., BHOSALE S., ET AL.: Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288 1* (2023). 2

[WAG22] WILES O., ALBUQUERQUE I., GOWAL S.: Discovering bugs in vision models using off-the-shelf image generation and captioning. In *NeurIPS ML Safety Workshop* (2022). 2, 3, 10

[WHM19] WILSON B., HOFFMAN J., MORGENSTERN J.: Predictive inequity in object detection. *arXiv preprint arXiv:1902.11097* (2019). 1

[XDL*24] XUAN X., DENG Z., LIN H.-T., KONG Z., MA K.-L.: SUNY: A visual interpretation framework for convolutional neural networks from a necessary and sufficient perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), pp. 8371–8376. 2

[XDLM24] XUAN X., DENG Z., LIN H.-T., MA K.-L.: SLIM: Spuriousness mitigation with minimal human annotations. In *European Conference on Computer Vision* (2024), Springer, pp. 215–231. 2

[XOG*25] XUAN X., ONO J. P., GOU L., MA K.-L., REN L.: AttributionScanner: A visual analytics system for model validation with metadata-free slice finding. *IEEE Transactions on Visualization and Computer Graphics* (2025). 2, 10

[ZCH*23] ZHU D., CHEN J., HAYDAROV K., SHEN X., ZHANG W., ELHOSEINY M.: ChatGPT asks, BLIP-2 answers: Automatic questioning towards enriched visual descriptions. *Transactions on Machine Learning Research* (2023). 5

[ZCS*23] ZOU Z., CHEN K., SHI Z., GUO Y., YE J.: Object detection in 20 years: A survey. *Proceedings of the IEEE 111*, 3 (2023), 257–276. 3

[ZCYT23] ZHANG Y., CARBALLO A., YANG H., TAKEDA K.: Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS Journal of Photogrammetry and Remote Sensing 196* (2023), 146–177. 1

[ZOS*22] ZHANG X., ONO J. P., SONG H., GOU L., MA K.-L., REN L.: SliceTeller: A data slice-driven approach for machine learning model validation. *IEEE Transactions on Visualization and Computer Graphics 29*, 1 (2022), 842–852. 1, 2

[ZPOH*24] ZHANG X., PIAZENTIN ONO J. H., HE W., GOU L., SACHAN M., MA K.-L., REN L.: Slicing, chatting, and refining: A concept-based approach for machine learning model validation with conceptslicer. In *Proceedings of the 29th International Conference on Intelligent User Interfaces* (2024), pp. 274–287. 1, 2, 3, 10

[ZZW*17] ZHU Y., ZHAO C., WANG J., ZHAO X., WU Y., LU H.: CoupleNet: Coupling global structure with local parts for object detection. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 4126–4134. 2, 4, 5, 10