Visual analytics towards detecting and explaining unknown-unknowns

Roy G. Biv, Ed Grimley, *Member, IEEE*, and Martha Stewart

Fig. 1. In the Clouds: Vancouver from Cypress Mountain. Note that the teaser may not be wider than the abstract block.

Abstract—Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

Index Terms—Radiosity, global illumination, constant time

1 Introduction

Artificial intelligence has been the driving force in automated problem solving in various domains. In general, such advances are driven by the evolution of computation hardware and data scale: models with a large number of parameters and layers, especially for deep neural models, are being developed to learn patterns in complex problems; it is supported by datasets with huge amounts of images and texts that were collected to provide possible real-world cases. As it has been increasingly adopted in high-stakes decisions such as medical diagnosis, self-driving cars, etc, it becomes imperative to cope with model failure: even a trivial model failure may lead to disastrous consequences. Yet, we find that there is still a lack of support on understanding why and how it fails. In this paper, we especially pay attention to a relatively little-known type of false predictions and resultant model behaviors called unknown-unknowns - when a model is trained on a dataset that is missing or biased against certain patterns thus does not well represent the real-world patterns, then it leads to unmodeled bias and exhibits a behavior of producing overconfident misclassification.

Motivating example. Take the well-known example of classifying images as dogs or cats in explaining unknown-unknowns when we train a model with a training dataset that contains color-biased cases including black dogs and non-black cats. As shown in the Fig. 2a, the model produces false predictions (caused by missing patterns of non-black dogs and black cats) when deployed in the wild. This problem is known to result from the discrepancy between the distribution of training set and out-of-sample set, which is referred to as several terms as well - domain shift or out-of-distribution.

Then, how can one tackle such problems to guarantee that the training data is diverse and complete enough to prevent such failure? In other words, how can we detect unknown-unknowns, reason about its rooted cause, and mitigate it? We find that there are some essentials along this way. In order to tackle it, the effective and valid approaches (1) *should be interpretable*: In case of image or text classification, which humans can reason about, judging whether a dataset has missing patterns is expected to be in a semantic way. For example, possible lacking patterns in the motivating example (e.g., missing white-colored dogs in dog class) are concepts which involve a segment (a set of pixels) or phrase (a set of words) that can be understood by human and hold as a part of real-world patterns, rather than arbitrary parameters with a single pixel or token, (2) *should hold deductive evidence*: Although we detect unknown-unknowns and confirm that they have certain patterns, it is imperative that we confirm whether the observed patterns actually caused false prediction. Going back to the misclassified case of a black cat in Fig. 1, we may ask, “Can we be sure that the black cat was misclassified as dogs because of blackness?” One may not know if it was due to other patterns (e.g., black-colored background or other object rather than blackness of the cat itself) until confirming what was learnt by model. Just identifying grouped patterns with an unsuper-

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

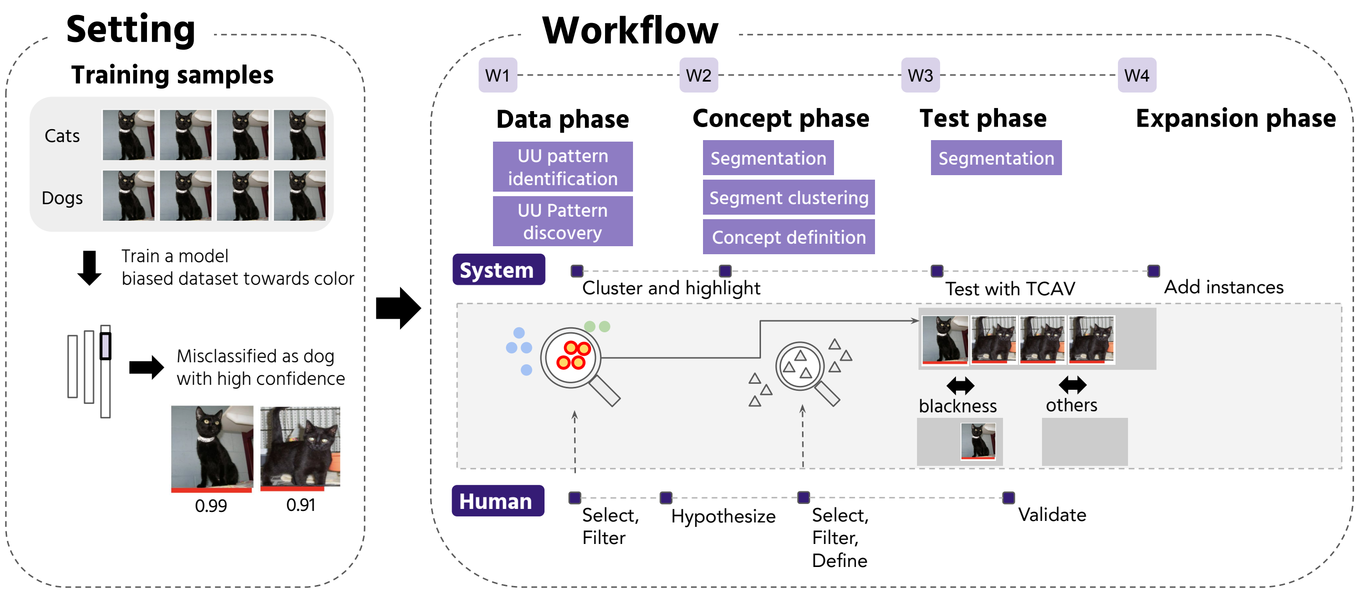

Fig. 2. **Workflow.** in-progress.

vised approach as proposed by [] may not provide explanations on why, therefore, hypothesized concepts need to be defined and tested. (3) *needs to be in a global manner*: we observe that an effective approach may lie in exploring global interpretability. To identify the rationale behind certain UU cases (e.g., why the model was misclassified black cats as dogs?), as mentioned above, the important task is to test how a semantic concept (e.g., blackness) was sensitive to the misclassified cats. It involves the task of collecting and defining semantic evidence (e.g., blackness represented by a set of black segments), and testing it to a set of instances that represent a subclass or entire class.

In the real-world practice, despite advanced data mining methods that were developed to identify unknown-unknowns, it never comes as an easy task; such process tends to bring noises that may not be in our interest. For example, in the image classification problem, setting and testing hypotheses on a semantic concept may require segmentation, clustering, and concept definition, but it is inevitable to encounter noisy segments or clusters. Most of segments may include no certain patterns and comprise clusters. We may want to explore some unknown-unknowns of our interest, however, selecting and testing specific unknown-unknowns cases (rather than whole class) is an iterative but hard task without an interactive agent with automated analysis support.

How can the tool help overview, detect, and reason the possibility of UU for datasets and models? To provide the systematic support in diagnosing unknown-unknowns problems, we propose a visual analytic approach for detecting and explaining UUs. Focusing on the deep image classification problem, we present a suite of visual components and interactions, ranging from model diagnosis and overview, to UU identification and concept testing. We define the workflow of these steps with three stages, which guide and inform users with similar analytic purpose. In concept testing, we leverage TCAV, the state-of-art interpretability technique, which fits well in our problem scope of UU identification and validation in a global manner. We bring it into practice in the analysis of UU with human-in-the-loop support, combining it with interactive segmentation, concept extraction and testing against a subclass of interest.

Our contributions include: 1) A visual analytic tool: We propose UUExplainer, a visual analytic toolkit for identifying and explaining unknown-unknowns. This is the first visual analytic toolkit that is tailored to dealing with UU problems. 2) Confusion matrix visualization and projection method: We propose a projection method in the visual space of confusion matrix to effectively reveal and contrast the patterns and UU cases in the instances. 3) A suite of evaluations: We present the use case scenarios and experiments to demonstrate the utility of our system and analysis.

2 Related Work

2.1 Detecting unknown-unknowns

The concept of unknown-unknowns has been introduced by [1] which had opened the discussion of its potential as one type of critical model failures. Introduced in the context of hate speech detection, which are also referred to as one of "Beat the machine" tasks, the problem is that the critical cases (i.e., examples of hate speech) are rare and varied, thus it results in the difficulty of letting a classifier learn diverse patterns of language uses. To mitigate this problem, they propose to leverage human ability to beat the machine, by letting them in reporting the misclassification and correcting the predictive tasks. The following studies in this line of work [4, 6] have also paid attention to human's such capability but with the support of automated analysis. These hybrid approaches propose the data mining method of classifying the examples of critical class [6], or detecting the UU cases with space partitioning [4]. Humans in this approach still play a major role, which machine is not better at doing, in annotating and finding out missing patterns that are complex and semantic with the aid of machine's preliminary analysis.

Fully automated analysis has been also introduced focusing on UU detection [2, 3]. They are motivated from the observation that such UUs tend to appear due to systematic biases in the training data. It is called patterned unknown-unknowns, which are known to form blind spots in the instance space. They propose to query such patterns with clustering techniques to isolate and detect the region of UUs with an efficient search with limited budgets [3] or a detection method ensuring large coverage to detect diverse patterns [2].

While existing literature contributes to mitigating UUs in varying directions with human or machine abilities, there is no endeavor to propose a systematic approach to guide crowdsourcing workers or data practitioners with visual cues. Considering the nature of tasks explored by previous studies that human-machine collaboration is effective in finding out missing patterns, we find that an interactive system is in need of supporting human-in-the-loop to intervene the automated analysis with the human guidance. In our study, we provide a workflow of detecting and explaining UUs in practice with our interactive tool to bring the aforementioned UU detection approaches in practice.

3 WORKFLOW

We present the workflow of how data practitioners can detect and explain unknown-unknowns with the help of a tool. Our goal is to provide the pipeline which consists of the realistic tasks with the goal of , but solve the obstacles by human-in-the-loop tool. Starting from the motivating example, think of an image classification task which learns patterns from a biased dataset. To train a model f:x -¿ y,

W1. Data phase. Identify the dominant patterns of unknown-unknowns.
In this phase, we aim to detect the candidate instances of unknown-unknowns. Following the observation from existing literature that UUs tend to form blind spots with regard to a missing pattern in the training set, our workflow involves clustering of instance activations to see which patterns model learned. To do so, UUs needs to be identified, selected, and explored to understand which visual patterns the model failed to predict.

T1. Detect the candidates of unknown-unknowns and clustered patterns

T2. Select, filter, expand instances to test

W2. Concept phase 1. Extract UU-related semantic concepts.
The concept phase is to capture solid evidence of what specific sub-pattern UUs are attributed to. For example, one of possible UU patterns in the motivating example is likely to include black cats because of the training set with a lack of such patterns. On the other hand, the model would have learned other sub-patterns to make predictions because cat images may have multiple sub-patterns within them, for example, ones that commonly appear in those images (e.g., cage or other objects) or any adversarial patterns. To provide evidence that is interpretable, we take the approach of explaining it with global interoperability, that is, how a concept (i.e., segment as a set of instances, rather than a feature (i.e., pixel as a single feature) is attributed to a set of instances. A concept in this context is defined as a coherent and semantic object or pattern in the dataset, for example, blackness or paws as shown in Figure x. As the dataset is a realization of such concepts in images, a concept (e.g., blackness) is equivalent to a set of segments that includes the corresponding pattern (e.g., black-colored patches). To operationalize this pipeline, we leverage TCAV, a global interpretability that is a model-agnostic explanation for deep classifiers. The final deliverable of this method is the score of how sensitive the model's prediction towards a class is to the concept. In detecting UUs, we extend the applicability of the method that any concepts can be tested towards a cluster as a specific pattern, or a set of UUs as a subclass or subset of instances.

We observe that, however, defining concepts is not a trivial task without an interactive tool. When users explore UU patterns, they may want to test and validate the sub-patterns within those images but it is likely that there are many noisy segments that are not of interest to users. The workflow supports selecting images of interest, doing segmentation of selected images, and clustering them. By selecting segments that are coherent to a concept of interest, a concept can be defined.

T3. Generate segments to find coherent concepts

T4. Define concepts with a set of segments

W3. Concept phase 2. Validate unknown-unknowns with concept hypothesis testing.
Once a concept is defined, the following process is to test and validate the concept by hypothesis testing.

T5. Test concepts against instances to explain the UU-relatedness

W4. Expansion phase. Expand the classifier with additional instances.

T6. Find similar instances in-the-wild

4 SYSTEM

UUExplainer is designed to operationalize the aforementioned workflow of explaining and detecting UUs. As shown in the Fig. x, the system begins by loading pre-trained models, which are the variants of deep image classifiers in . To facilitate UU detection, which serves as the first step of the workflow, Confusion Space and Instance Viewer gives the overview of distinct patterns with respect to UU. The instances can segmented to define concepts via Segment View and , or be selected to test its sensitivity towards defined concepts. All the experiment results conducted by users are listed up in the . We provide the detailed description of visual encodings, methods and interactions for each analytic component.

4.1 Model Viewer

4.2 Confusion Space

Confusion Space allows to inspect the classification result of two classes by the selection from Confusion Summary, which is represented as the four cases of confusion matrix. With four 2d plot in the 2x2 layout for each corresponding to a confusion case (TP, TN, FP, FN), test instances that belong to each region are plotted.

Specifically, Confusion Space serves two purposes: 1) identifying patterns, 2) detecting out-of-distribution regions. As introduced earlier that UUs may be rooted from systmatic biases with certain patterns missing in the training phase, we provide the visual overview of patterns. In order to give a visual overview of how images are grouped together by their patterns, we first conduct a clustering to identify the clusters of instances, then employ a cluster-constraint dimension reduction to force the coordinates of instance circles to be partially guided by their grouped patterns (from the clustering result). In those tasks, we conduct k-means clustering with activations of test instances X from a layer. For the purpose of visualizing instances in the 2d plot in terms of not only their similarity but clustering result, we conduct supervised t-sne [5] that Each instance is rendered as a circle with its coordinate along x and y-axis representing the result of 2d projection. In this task, we leverage the using the t-sne technique.

4.3 Instance Viewer

Instance Viewer provides the list of image snippets for a closer inspection of selected images. Connecting to the layout of Confusion Space, it displays the original images of instances with respect to four confusion cases that are horizontally aligned. To facilitate the instant overview, images with higher with a red bar on the bottom of each image as an indicator, or selected instances with its blue stroke are sorted and displayed in the front.

4.4 Focused View

Focused View gives an overview of selected images from Confusion Space. allows users to contrast the instances that are similar , different cases highlighted in the selected region. Once users select a region of interest from Confusion Space, it highlights the instances within the selected region of four confusion spaces. As a result, users can expect that those instances are the ones that have similar patterns that are closely plotted, but in a different class or prediction. ...

4.5 Segment View

Segment View supports conducting segmentation and selecting a set of coherent segments to define a concept as an intermediate step to automate the concept definition and testing. When instances are selected, users can generate segments from those selected images by clicking the segmentation button, which activates the segmentation method called slic. While the segmentation task is conducted with the original pixel information of images, we represent the segments with their internal representation of the target layer to understand how model learned such subpatterns. In order to visually reveal the segments, `Segment View` clusters and reduces the dimension of segment representation to plot them in two dimensional plot, representing them as circles with its color indicating cluster membership.

Users are allowed to select a set of segments to define a concept. Once a segment circle is hovered, the nearest 20 segments in terms of euclidean distance are displayed in the tooltip to allow users to check the visual coherency. When the circle is clicked, those 20 segments are selected together and defined as a concept. Any defined concepts are listed in the concept list placed right bottom of `Segment View`.

4.6 Experiment List

$\sum_{i=1}^{N} \mathbb{1}(TCAV_i > TCAV_c)$
$\sum_{i=1}^{N} UU_i * TCAV_i$

Acknowledgments

The authors wish to thank A, B, and C. This work was supported in part by a grant from XYZ (# 12345-67890).

References

[1] J. Attenberg, P. Ipeirotis, and F. Provost. Beat the Machine: Challenging Humans to Find a Predictive Model's "Unknown Unknowns". *Journal of Data and Information Quality*, 6(1):1–17, Mar. 2015. doi: 10.1145/2700832

[2] G. Bansal and D. S. Weld. A Coverage-Based Utility Model for Identifying Unknown Unknowns. p. 8.

[3] H. Lakkaraju, E. Kamar, R. Caruana, and E. Horvitz. Identifying Unknown Unknowns in the Open World: Representations and Policies for Guided Exploration. p. 9.

[4] A. Liu, S. Guerra, I. Fung, G. Matute, E. Kamar, and W. Lasecki. Towards Hybrid Human-AI Workflows for Unknown Unknown Detection. In *Proceedings of The Web Conference 2020*, pp. 2432–2442. ACM, Taipei Taiwan, Apr. 2020. doi: 10.1145/3366423.3380306

[5] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[6] C. Vandenhof and E. Law. Contradict the Machine: A Hybrid Approach to Identifying Unknown Unknowns. p. 3, 2019.