TU Delft

Delft University of Technology

What do You Mean? Interpreting Image Classification with Crowdsourced Concept Extraction and Analysis

Balayn, Agathe; Soilis, Panagiotis; Lofi, Christoph; Yang, Jie; Bozzon, Alessandro

DOI
10.1145/3442381.3450069

Publication date
2021

Document Version
Final published version

Published in
The Web Conference 2021 - Proceedings of the World Wide Web Conference, WWW 2021

Citation (APA)
Balayn, A., Soilis, P., Lofi, C., Yang, J., & Bozzon, A. (2021). What do You Mean? Interpreting Image Classification with Crowdsourced Concept Extraction and Analysis. In *The Web Conference 2021 - Proceedings of the World Wide Web Conference, WWW 2021* (pp. 1937-1948). (The Web Conference 2021 - Proceedings of the World Wide Web Conference, WWW 2021). https://doi.org/10.1145/3442381.3450069

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

What do You Mean? Interpreting Image Classification with Crowdsourced Concept Extraction and Analysis

Agathe Balayn, Panagiotis Soilis, Christoph Lofi, Jie Yang, Alessandro Bozzon
{a.m.a.balayn;c.lofi;j.yang-3;a.bozzon}@tudelft.nl;panagiotis.soilis@gmail.com
Delft University of Technology
Delft, The Netherlands

ABSTRACT

Global interpretability is a vital requirement for image classification applications. Existing interpretability methods mainly explain a model behavior by identifying salient image patches, which require manual efforts from users to make sense of, and also do not typically support model validation with questions that investigate multiple visual concepts. In this paper, we introduce a scalable human-in-the-loop approach for global interpretability. Salient image areas identified by local interpretability methods are annotated with semantic concepts, which are then aggregated into a tabular representation of images to facilitate automatic statistical analysis of model behavior. We show that this approach answers interpretability needs for both model validation and exploration, and provides semantically more diverse, informative, and relevant explanations while still allowing for scalable and cost-efficient execution.

CCS CONCEPTS

• **Information systems → Crowdsourcing**; • **Human-centered computing → Collaborative and social computing**.

KEYWORDS

machine-learning interpretability, image classification, human computation, concept extraction

ACM Reference Format:
Agathe Balayn, Panagiotis Soilis, Christoph Lofi, Jie Yang, Alessandro Bozzon. 2021. What do You Mean? Interpreting Image Classification with Crowdsourced Concept Extraction and Analysis. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3450069

1 INTRODUCTION

State-of-the-art image classification methods employ neural models, generally operating as "black-boxes". The opaqueness of these models has become a major obstacle for deploying, debugging, and tuning them [11, 24]; particularly in critical domains such as health, security and justice, where the ability to understand, or, at least, to *interpret* their behaviour is increasingly demanded [21, 35].

Interpretability in machine learning refers to "the ability to explain or to present in understandable terms to a human" [11] how the model makes a prediction. It is critical to understand and improve model performance, and to establish user trust in the model and its behaviour. To be effective, interpretability methods must: (**1**) present interpretations that match humans' mental representations of *concepts* [2, 19, 22] as humans understand the world through concepts associated with observable properties. Human brains process visual information from low-level concepts such as color, contrast, to mid-level ones such as shapes, textures, and to more abstract semantic representations of an object. For example, an *ambulance* is "a car-shaped object that has a red cross or blue star symbol on it". And, (**2**) allow for the satisfaction of *interpretation needs* aimed at both model behavior *validation* and *exploration*.

A typical validation scenario occurs when a model developer (or auditor) tests precise hypotheses on the workings of automated decision making to ensure the system behaves as intended. In an ambulance recognition example (Figure 1), an auditor could ask "In the classification of ambulances, does the model focus on the `red cross and the flash lights`; or does it focus on unrelated background concepts like the `blue sky`?". In an exploratory scenario, the developer would be interested in understanding the classification behaviour of the model, but without a precise hypothesis to test. To support both scenarios, an interpretability method should be able to test for the *presence*, *combination*, or *absence* of *multiple concepts* with varying granularity –e.g. a model might learn to use an ambulance's `overall shape` (coarser granularity), or the `sign` on the frame `and the flash light` (finer granularity).

Despite the recent advances in interpretable machine learning [6, 13, 15, 24, 39], existing methods addressing image classification fall short in meeting the above requirements. We focus on *post-hoc* interpretability methods, which, in contrast to *inherent* interpretability methods (see a detailed discussion in Section 2), can be applied to any existing classification model. Among post-hoc methods, *global interpretability* methods [13, 15] support exploration needs by automatically producing "patches" from multiple images in the dataset (ACE [13]) that should represent one visual concept inferred to be important for classification; or, for validation purposes, require users to provide a set of images (patches) as examples of visual concepts (TCAV [15]). Both approaches have shortcomings. First, they require manual analysis and interpretation to associate image patches with understandable concepts and properties [13], or require an input set of example images that cleanly capture the interpretation hypothesis the user wants to verify (e.g., images of ambulance with a cross sign but without a sky background) [15]. Besides, such methods do not easily support the validation and exploration of *multi-concept* interpretation. On the other hand, *local interpretability* methods analyse individual images [29, 43] and produce image-specific *saliency maps*, i.e. a highlight of the most

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.
WWW '21, April 19–23, 2021, Ljubljana, Slovenia
© 2021 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.
ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3450069

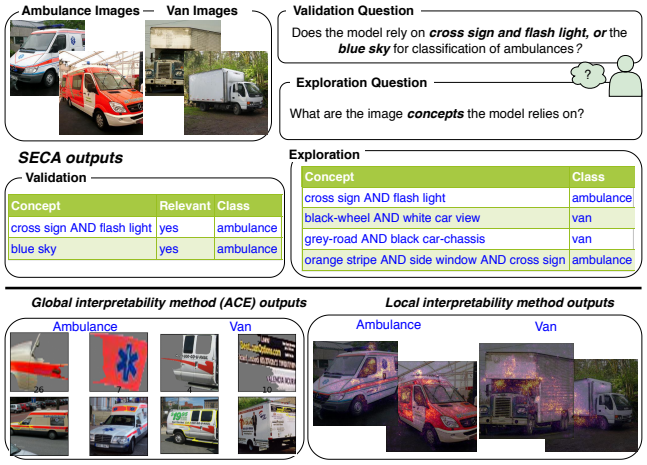


Figure 1: SECA generates (multi-concept) interpretations for both model behavior validation and exploration. In contrast, state-of-the-art *global* (e.g., ACE [13]) or *local* interpretability methods do not support multi-concept interpetability need, and generate image patches or saliency maps (for exploration only) that require manual interpretation.

important pixels for the classification of a given image. Local methods can be adopted for global interpretability, but with significant cognitive demand on users, both for validation and exploratory interpretation needs: multiple images must be individually analysed to associate image regions with intelligible concepts, and the respective concepts need to be reconciled globally and interpreted against the classification behaviour of the model.

Arguably, a better interpretability method would combine the ability to analyse classes of images and support multi-concept interpretation for both model validation and exploration purposes without imposing high cognitive load to its users to make sense of interpretation outputs. With this in mind, we designed SECA, a human-in-the-loop SEmantic Concept extraction and Analysis framework that supports global analysis of machine behavior for multi-concept questions. SECA generates interpretation with a rich set of semantic concepts easily comprehensible by users. It fuses local interpretability methods to identify image patches that are relevant to the prediction for individual images, with human computation to annotate those patches with *semantic concepts*, i.e., visual entities with types and attributes. Using the entities, it then builds a model-agnostic structured representation of dataset images, on which statistical analysis techniques can be applied to answer both validation and exploratory interpretability questions. The combination of local interpretability methods, crowdsourcing, and statistical analysis techniques allows for scalable extraction and analysis of relevant concepts from a large number of images to facilitate validation and exploration of a model's behavior.

We demonstrate the *correctness*, *informativeness*, and *effectiveness* of SECA through several interpretability scenarios and evaluation protocols. To deal with the lack of ground truth of model behavior (a common issue in interpretability literature [11]), we design controlled experiments where several types of pre-defined model biases are induced, ranging from simple visual entities to complex ones related to image scene understanding. We conduct empirical studies to understand the cost/effectiveness trade-off with varying number of images, granularity of annotations, and crowd involvement. In summary, we make the following key contributions:

- A novel human-in-the-loop interpretability framework that allows for statistical analysis of global model behavior through rich multi-concept interpretability questions.
- A benchmark for evaluating global interpretability methods for multi-concept questions, including interpretability scenarios across three image classification tasks with different types of biases.
- An extensive evaluation of the framework, demonstrating its effectiveness for both model validation and exploration, and analyzing its configurations for optimal cost/effectiveness trade-off.

A replication package containing code, datasets, and unabridged experimental results is available on the companion page[1].

2 RELATED WORK

We first provide an overview of existing interpretability methods, then focus on approaches specific to image classification, and finally discuss works on human-in-the-loop machine learning.

2.1 Machine Learning Interpretability

Existing interpretability methods can be categorized in two ways: i) *local* vs. *global*, depending on the scope of data instances interpreted being individual instances or class of instances; or ii) *post-hoc* vs. *inherent* interpretability methods, depending on whether the goal is to provide interpretations for an existing model or constructing self-explanatory models. Inherent interpretability is achieved by adding interpretability constraints in model learning to enforce feature sparsity [12], representation disentanglement [44], or sensitivity towards input features [33]. Another popular approach is attention mechanisms, which identify parts of the input that are attended by the model for specific predictions [7, 37]. Turning an existing model into an inherently interpretable model might be costly for users and might lead to a drop of model performance. In contrast, post-hoc interpretability methods can be applied without model modification or retraining, and have therefore attracted growing attention. Our SECA is a post-hoc interpretability method.

A key challenge in post-hoc interpretability is interpretation *fidelity*, i.e., ensuring that the generated interpretation accurately describes model behavior. This can be achieved in several ways. Koh and Liang [16] propose a perturbation-based method that identifies training instances most responsible for a given prediction through influence functions, which estimate changes in model parameters as an effect of changes in the training instances. Gradient-based methods calculate the gradient of the output with respect to the input to derive the contribution of features [5, 25, 29]. Ribeiro et al. [24] fit a simpler model (with interpretable features) around the test instance to ensure local consistency between the interpretation and model prediction. A simple interpretable surrogate model can be learned to approximate the original model's predictions on a representative sample of the data [32]. Our approach is inspired from this last idea, as it generates interpretations using statistical

[1] https://sites.google.com/view/webconf21-whatdoyoumean-balayn

tools such as association rule mining and decision trees (on human intelligible concepts) that are self-explanatory.

2.2 Interpreting Image Classification

The most extensively studied interpretability approach for image classification is *saliency*, a local interpretability post-hoc method that highlights the most important pixels of an image for model decisions in what is called a saliency map [29] . "Importance" is defined as the sensitivity of decisions to the pixels with respect to a specific class. It is measured either by computing the gradient of the activation function for that class with respect to every image pixel [27, 29], or by passing the activated features of each layer of the model backwards into a reverse neural network model until the activations are mapped to the actual inputs of the model [6, 28]. Those approaches are likely to generate noisy results highlighting irrelevant pixels. To deal with that, methods such as SmoothGrad [31] and the Integrated Gradient [33] have been proposed.

Due to the intrinsic lack of semantics in pixels, global interpretability is challenging in image classification. Kim et al. [15] introduce TCAV on top of their notion of Concept Activation Vectors (CAVs), which represents the translation from the internal states of a model to human-understandable concepts. The importance of a concept for model predictions is measured by calculating the directional derivative w.r.t. the corresponding CAV, i.e., the sensitivity of model predictions to changes in inputs towards the direction of the concept. A main disadvantage of such an approach is that CAVs are obtained by training a linear classifier between a concept's examples and counterexamples; as a requirement, users need to provide sets of (50-150) example images for the training. Such a process is not only expensive, but sometimes also infeasible when the concept for testing comprises multiple concepts: users need to prepare a number of example images that each cleanly captures the multiple concepts that the user wants to verify. Moreover, the method is designed for model behavior validation; exploratory analysis is possible, but clearly expensive.

Ghorbani et al. [13] introduce ACE to automatically extract visual concepts, by aggregating related local image segments across the data. It relies on automatic image segmentation and clustering to obtain image patches potentially representing the same concept, and then uses TCAV to test for its importance. The quality of generated interpretations is highly dependent on the effectiveness of image segmentation and clustering: our experiment shows that ACE is prone to identify patches representing a concept related to low-level visual information (e.g., color), and that it fails at identifying patches of concepts comprising multiple concepts (Section 5.2 and 5.3). What is more, image patches generated by TCAV are not self-explanatory, and need to be analysed and interpreted by users.

By a combination of local interpretability and crowdsourcing techniques, the SECA framework can address both issues of fidelity and cognitive load by 1) relying on human annotations to present semantic concepts at different conceptual granularities, and 2) by enabling multi-concept model validation and exploration.

2.3 Human-in-the-Loop Machine Learning

Human-in-the-loop machine learning [36] has been traditionally concerned with crowdsourced training data annotation [10] and crowd-collected samples [8]. A closely related line of work is "learning from crowds", where researchers study models that can learn from noisy crowd labels [23]. Unlike the conventional learning setting, these models are concerned with learning parameters of the annotation process (e.g., annotator expertise, task difficulty) and inferring true labels from noisy ones, possibly by incorporating (deep) active learning to reduce annotation efforts [38, 40].

Recent works address the use of human computation to debug machine learning systems. Nushi et al. [20] use crowdsourcing to identify weakest components of a machine learning pipeline and to propose targeted fixes. Yang et al. [41] introduce a human-in-the-loop system for debugging noisy training data using an automatic method for inferring true labels and crowdsourcing for manual correction of wrong labels. Hu et al. [14] introduce a crowdsourcing workflow for detecting sampling biases in image datasets.

The use of human intelligence for interpreting machine learning models has been limited to involving humans as users for evaluating the interpretability methods, e.g., by observing if the interpretations help users choose a better model [11, 24]. Unlike those methods, SECA involves human computation as an integral component to identify relevant concepts, which is of crucial importance to make interpretations intelligible and to support multi-concept queries.

3 DESIGN PRINCIPLES AND CHOICES

We design SECA with the following key requirements in mind: (**1**) *Intelligibility*, the generated interpretation output should be comprehensible by its users; (**2**) *Effortlessness*, the cognitive load imposed on users should be minimal; (**3**) *Utility*, the framework should support both confirmatory or exploratory questions for model validation and exploration; (**4**) *Fidelity*, the generated interpretation should correctly and comprehensively describe the model behavior; (**5**) *Scalability* and *cost-effectiveness*, the framework should be scalable and effective under reasonable cost. In the following, we describe our design choices following from each of the above requirements.

3.1 Intelligibility

To cater for *intelligibility*, we draw inspirations from the cognitive psychology literature on human reasoning and concept creation. Aerts [2] considers that *concepts* can be associated with observable properties, and the degree of association, called *typicality*, can be measured, typically by asking humans to rate it on a Likert scale. For instance, the concept `ambulance` can be associated with the property `cross sign`. Clearly, a property could be a concept itself, or be composed of multiple concepts [3]. The Representational Theory of Mind proposes a compositional semantic [19], where two or more "noun" concepts, or "noun" and "adjective" concepts can be combined using syntactic rules.

In this work, we consider *interpretability needs* aimed at analysing the degree of association (*typicality scores*) between *concepts* appearing in images (e.g. `cross sign`) and the classification *labels* –also concepts (e.g. `ambulance`)– that a machine learning model assigns to them. Those concepts correspond to *entity types* (nouns, e.g. `cross sign`) or *entity attributes* (adjectives, e.g. `red`) drawn from a vocabulary. *Interpretability needs* are expressed as *textual* queries over concepts, possibly using logical operations –conjunction (`AND`),

disjunction (OR), and negation (NOT). An example of query (section 5) is: "orange-stripe AND light AND NOT chassis".

3.2 Utility and Effortlessness

We represent images and classification labels through the list of concepts, i.e. *entity types* and *attributes* they contain. Without loss of generality, in the following we consider only classification labels related to a single concept (e.g. male/ female). We only consider a binary representation of a concept's relation to an image (presence/absence of a concept); a weighted representation (e.g. a value between 0 and 1) is an extension that we leave to future work. By explicitly identifying concepts on a per-image basis, we can apply a set of statistical analysis tools to identify the importance of concepts (individual or combined) across images in relevance to model predictions. This lessens the users cognitive load –many other global interpretation approaches rely on human user to identify relevant concepts across several images–, and allows to investigate more diverse model behavior.

3.3 Fidelity and Scalability

To ensure interpretation fidelity, we use only *relevant* concepts. To do so, we rely on existing local interpretability methods: we compute the saliency maps for (a subset of) images on which a model makes predictions, and create semantic descriptions of the entity types and attributes in the areas highlighted in the maps.

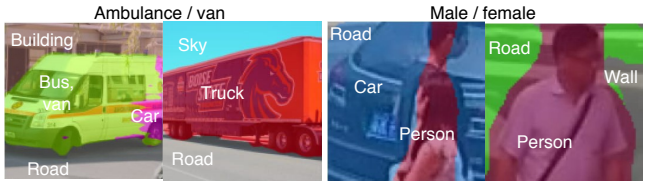


Figure 2: Automatic semantic segmentation with DeepLabv3. The truck and ambulance (left) appear as single segments while more specific entities like stripes and flash light are probably used by the model. The silhouettes of the individuals (right) form a single segment and the background another, whereas a model likely uses finer-grain entities (e.g. hair length, face shape).

This annotation process cannot currently be automated, as state-of-the art segmentation and object recognition methods are not accurate enough to uncover entities or attributes relevant to a model's decisions. In Figure 2 examples, the granularity of the segmented entities is large and the annotations vague. For instance, an ambulance is segmented as one entity and annotated as bus.

Hence, SECA adopts a crowdsourcing approach, where crowd annotators are asked to identify and describe with a textual annotation each entity in the salient image areas. Such approach can provide high fidelity and, while incurring some unavoidable costs, be scalable. Section 4 describes how SECA tackles obvious issues of annotation coherency across images. In the experiments of section 5 and section 6, we empirically study fidelity and cost-effectiveness, showing the quality and feasibility of the approach.

4 THE SECA FRAMEWORK

Figure 3 presents an overview of SECA (SEmantic Concept extraction and Analysis). Given as input (**1**) a trained image classification model and (**2**) a dataset, SECA can answer interpretability questions for validation and exploration purposes. (**C1**) Images in the dataset and their corresponding predicted labels are passed through a local interpretability method. The method generates saliency maps that indicate pixels relevant for the model prediction. (**C2**) All maps and corresponding images are sent to human annotators, to collect semantic annotations about the types and attributes of entities represented by the salient pixels. (**C3**) Annotations across images are reconciled, and (**C4**) a structured and consolidated representation of all images is built. Finally, (**C5**) data analysis tools are applied, and single and multi-entity concepts and their *typicality scores* (degree of association of the concept and a target label) are outputted.

C1: Saliency Map Extraction. Saliency maps extraction is necessary to provide accurate interpretations while reducing annotation effort: clearly, annotating an entire image would be more expensive, and it could introduce concepts that are not germane to a model's behaviour interpretation. SECA is agnostic to the employed local interpretability method. We opted for SmoothGrad [31], which is sensitive to the parameters of a model (thus catering for more accurate capturing of a model behaviour) while minimising noisy results (i.e., highlighting irrelevant pixels). To further reduce annotation efforts, saliency map extraction is performed only on a random sample of all images. An appropriate setting of the number of sampled images depends on the complexity of the machine learning task, e.g., number and diversity of relevant concepts. We study the quality/cost trade-off related to this number in section 6.

C2: Saliency Maps Annotation. The annotation task combines two typical crowdsourcing activities: drawing bounding boxes and labelling (parts of) images. We ask workers to (1) identify, for each salient pixel area, the *entity types* corresponding to recognizable object shapes, and the *entity attributes* characterizing the area, e.g., its colors, textures or object property; (2) draw bounding boxes around the pixels corresponding to these types and attributes (we use bounding boxes instead of continuous curves as it is easier and faster for crowd workers); (3) provide a textual description (one word) of the identified types and attributes. For example, if the saliency map focuses on the blue cross image area on the trunk of an ambulance, the annotation would be *type*: cross; *attributes*: blue; for a gender classification task, a saliency map focusing on a person's short black hair results in *type*: hair; *attributes*: black, short. Entity-attribute information per salient image area is relatively easy to create by annotators, relevant to interpretation (as they are based on saliency maps of model predictions), and naturally intelligible for model developers and auditors. We ask annotators to provide fine-grained annotations, as fine-granularity entities can be later aggregated. Automatic checks are implemented to ensure that each image has at least one bounding box, and each bounding box has at least one entity type and attribute annotated. We employ multiple crowd workers per task to maximize the number and diversity of relevant annotated concepts. We retain concepts annotated by workers who spend more than a pre-defined amount of time on each image. The annotation task design is available on the companion page. Parameters of the *C2* component that affect the

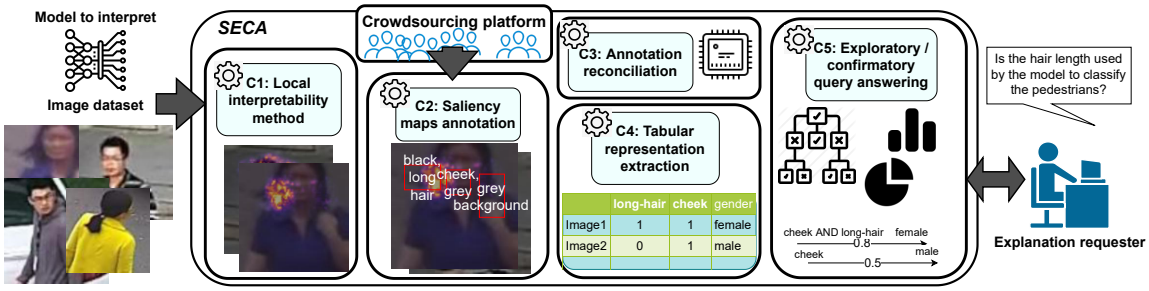


Figure 3: Overview of the SECA framework.

cost-effectiveness of SECA are *annotation granularity* and *annotator type* (e.g. experts vs crowds). We study their impact in section 6.

C3: Annotations Reconciliation. Annotation reconciliation is required as no pre-defined vocabulary of entity types and attributes is imposed on annotators, thus leading to diversity in vocabulary and/or granularity. First, we correct spelling mistakes with spell-checkers [2], normalize the annotations by removing white spaces and converting all characters to lowercase, and rename synonyms or highly similar annotations using a reconciled term. The reconciled term is obtained by automatically clustering all the collected terms represented by word embeddings (pre-trained FastText embeddings), and picking the one closest to the centroid of each cluster. We use K-mean clustering, where k is chosen by identifying the value that leads to distributions of Silhouette score per cluster that do not exhibit negative values and that are as much uniform as possible across clusters. Features for the tabular representation are then built by mapping each annotation to one cluster or the association of multiple clusters. E.g., `wheel` is associated to the cluster containing this term, while `front light` is associated to the super cluster that combines the clusters of `front` and `light`. Annotation errors should not propagate as we later retain only interpretations that are statistically significant. In future work, we plan to look into (dynamically) controlling for vocabulary in the annotation task.

C4: Tabular Image Representation. The reconciled annotations of the salient areas of each image are stored in a de-normalised form. We create a binary-value column for entity type-attribute combinations (like `hair-short-black`), but also columns for each component (`hair`, and `short`, and `black`). For each image, we store which entity types and attributes pairs have been connected to any of their salient pixel areas. This denormalized storage helps with further statistical analysis and querying: for instance, a user could investigate three hypotheses: is the `cross` logo indicative of ambulances identified by a model predictions? Are `orange crosses` even more relevant? Has the model learned to check solely for the `color orange` (strongly correlated with ambulances)? The entity type `cross` can address the first question, the pair `cross-orange` the second, and the attribute `orange` the third.

C5: Query Answering. This component generates interpretations to fulfill both interpretation needs of model validation and exploration. The interpretations take the form of tuples corresponding to *a*) a concept, *b*) a prediction label, and *c*) a typicality score that measures the importance of a concept in predicting the label by the model. The tuples are then ranked based on the typicality scores.

Statistical tools. The most relevant concepts to include in output are identified through *statistical tests* assessing the correlation between each concept (i.e. column) present in the tabular representation and the predicted labels. We use the Chi-Square independence test [46], to check whether a concept and the label are independent. We retain concepts that are not independent significantly (p-value < 0.05). We compute the *Cramer's V* test [1] (a test commonly used in interpetability literature) on the retained concepts to obtain a *typicality score* that measures their degree of association with the labels. We also perform a *frequency analysis* of each concept per class, to identify concepts relevant for multiple classes simultaneously.

To facilitate *exploratory* needs, we pre-compute combination of concepts as follows: for each concept found significant, we add to the tabular representation a column with the complementary of the original column of the concept –this encodes the `NOT` operator of the concept, i.e. its absence. We also add columns that encode the logical `AND` combination of concepts (e.g. if `wheel` and `light` are found significant, we append a `wheel AND light` column). We then repeat the process of computing the statistical tests to identify the significant concepts among these new columns. Obviously, it is possible to explore all possible combinations of concepts; without loss of generality, in this paper we limit to pairwise combinations.

For model *validation purposes*, users can query over the concepts present in the tabular representation, possibly using logical operators. If not existing, the query is translated into a new column encoding the queried (multi-entity) concept. Statistical tests are then applied to establish the significance of the new column.

Rule extraction tools. The set of concept combinations is extended through rule extraction methods, uncovering multi-entity concepts that involve more than one `AND` or `NOT` logical combination. We employ *association rule mining* algorithms and *decision tree* classifiers. Association rules provide indications on the co-occurrence relationships between concepts within the rules. We apply the Apriori algorithm [4] on the original tabular representation, and constrain it to generate rules where the rule bodies are image concepts and the rule heads are the prediction labels. We use the *lift* score (a measure of the importance of a rule) as the typicality score of the rules. Unlike association rules that only captures co-occurrence relations, rules extracted from decision trees [9] contain numerical threshold for each concept. We use accuracy and frequency of the rule as its *typicality scores*. Decision trees require sufficient training

[2] SymSpell: https://github.com/wolfgarbe/symspell

data to be employed, so their applicability is conditional to the amount of considered images, but their output is richer.

5 PERFORMANCE EVALUATION

We evaluate the interpretation performance of SECA by investigating two questions: *Q1: how correct are interpretations provided by* SECA *for uncovering biased behaviors?*, and *Q2: how informative are those interpretations in comparison to other interpretability methods?*

5.1 Experimental set-up

To date, no benchmark exists to measure the performance of interpretability methods for multi-concept questions. Inspired by previous evaluations [15], we design the following procedure.

5.1.1 Evaluation process. ***1)*** *Correctness.* We consider interpretations *correct* if they highlight the concepts used by a model to make its predictions. Correctness is assessed by comparing these interpretations to a ground truth in controlled experiments. As such ground truth is not readily available, we generate it by biasing the models' behavior, i.e. we force models to "focus" on certain types of concepts that are exclusive of different classes. We create this bias either by injecting visual entities into images (e.g. adding time stamps to each image of a selected class), or by re-sampling the dataset based on existing entities (e.g. making sure that all images of a class present an object from an angle different from images of other classes). We verify that the trained models learn these biases by computing the training accuracy: accuracy close to 1.0 indicates the models fit the data very well, probably thanks to the bias which is easy to pick up on. To further evaluate the correctness of SECA, we check its ability to highlight differences in "less obvious" (or less skewed) variations of model behaviors that are due to differently (less) biased composition of training datasets, or to the variations in the model architectures, under the assumption that these models should rely partly on different concepts to make their predictions. All these interpretability scenarios are summarized in Table 1.

2) *Informativeness.* Interpretations are informative if they uncover concepts that are *diverse* – presence of single and multi-entity concepts with various logical connections, and *actionable* for model debugging – concepts that show a potential issue and that are enough informative to act on them, e.g., by modifying the distributions of the corresponding visual entities in the training dataset.

5.1.2 Evaluation details. ***1)*** *Learning tasks.* We select three classification tasks from two popular datasets for computer vision benchmarking: a *gender classification task* (T1) from pedestrian images using the PA-100K dataset [17][3]; a *three-class "fish" classification task* (T2) containing lobster, great white shark and tench images; a *two-class vehicle classification task* (T3) with moving van and ambulance images from the ImageNet ILSVRC-2012 dataset [26].[4] We crop and rescale the dataset images to input them to the machine learning models. We balance the data for equal representation of the classes (49000 images for T1, 4500 for T2, 3000 for T3).[5]

2) *Machine learning models.* We experiment with Inception V3 [34](M1) and VGG16 [30](M2), both pre-trained on ImageNet, and fine-tuned on the evaluation datasets. Those models were shown to learn different feature representations [45].

3) *Bias injection in Data.* Inspired by Yang and Kim [42], from the PA-100K we create 4 experimental datasets by injecting text as visual entities into the pedestrian task data: *Date dataset* (D1): date stamps on the `female` images and datetime stamps on the `male` ones – the model should rely on the presence or absence of the entity type `time stamp`; *Color dataset* (D2): `white` and `yellow` dates respectively on the `female` and `male` images –the model should rely on the `white` and/or `yellow` color attributes; *Date City dataset* (D3): `date`, or `datetime` and `city name` in the `female` images, `datetime`, or `date` and `city name` in the `male` images –the model should rely on combinations of entity types; *Colored-Date dataset* (D4): `white dates` or `yellow datetimes` in the `female` images, and `yellow dates` or `white datetimes` in the `male` images –the model should rely on pairs of color and entity types. In *Orientation dataset* (D5.2), we resample images of PA-100K (D5.1) by imposing a class-specific pedestrian `orientation` – all `male` images have a `front orientation` (i.e. the pedestrian face is seen), and all `female` images a `back orientation`. Models trained on it should learn concepts characterizing the front and back of a person. These datasets should bias the model towards diverse concepts based on different entity types, attributes and their combinations, exactly what an interpretability method should uncover.

4) *Bias injection in Model Architectures.* We create different model behaviors to compare by using the pre-trained models to make predictions on the fish (BM1.1) and vehicle tasks (BM2.1), and by fine-tuning these models solely on the target classes of these tasks (i.e. training the models further only with the data of these classes) (BM1.2, BM2.2). Fine-tuning should bias the behaviors towards background concepts as these classes bear strong skew towards background entities (e.g. sharks are almost all in the ocean, tench with a fisherman next to a forest or grass, lobsters on a plate).

5) *Baseline.* We compare SECA interpretations to the only automatic interpretation approach in literature, ACE [13]. We do not consider TCAV [15] because it requires input "query" concepts. The study on the relationship between input patches and interpretation performance is beyond the scope of this paper. ACE outputs sets of 10 image patches, that should be interpreted by the user as single concepts. We retain ACE's sets that have a p-value under 0.05. It is generally difficult to associate meaningful semantic concepts to the sets, because their patches contain different entity types, thus making the underlying concept hard to identify. E.g., the underlying concept for image patches of `grey water`, `grey shark fin`, and `grey shark stomach` is ambiguous (could be the grey color and/or shark body parts)[6]. We retain *recognizable* visual concepts that are present at least in 5 of the 10 example patches of a set.

6) *Annotation of Saliency Maps.* To avoid confounding factors from crowd work ambiguity, in these experiments trained annotators (the authors) annotated the saliency maps, with agreement reached on the fine concept granularity. After experimenting on the learning tasks, we set $\sigma = 5$, $n = 10$ for SmoothGrad. For every task, the annotators annotated 300 images – as detailed in section 6, this amount is sufficient to cover concepts relevant to model behavior.

[3] We acknowledge the limitations of a binary gender, but no other dataset was found.
[4] This task is inspired from [18] that hints at biases in background of these images.
[5] Our pre-processed dataset will be made available upon acceptance of the paper.
[6] The companion page reports highly ranked non-recognizable concepts from ACE.

Table 1: Summary of the interpretation scenarios.

Task	Bias injection
T1: gender	D1-D4: text and color visual entities D5.1 / D5.2: original data / orientation bias
T2: fish	BM1.1 / BM1.2: original data / fine-tuned model
T3: vehicle	BM2.1 / BM2.2: original data / fine-tuned model
ML model	M1 / M2: Inception V3 / VGG16

Table 2: Example interpretations of SECA on the pedestrian classification task with simple injected biases.

Bias type	Output interpretations (rank - Cramer's value)
date (D1)	hour, NOT hour, minute, NOT minute (1-.93), hour AND minute(2-.9), day AND minute(4-.47), day(10-.24)
color (D2)	yellow-year(1-.96), yellow(2-.94), white(3-.83), yellow-day(4-.82), yellow-month(5-.81), white-year(6-.72)
date city (D3)	NOT city AND NOT minute(1-.5), NOT city AND NOT hour(2-.49), city AND NOT hour(3-.46), city AND hour(4-.45)
colored date (D4)	yellow-hour(1-.6), yellow-minute(1-.6), white-minute(2-.53), white-hour(3-.52), yellow-day AND yellow-year(4-.37)

5.2 Results: Correctness

In the following tables, we only report the simple and multi-entity concepts that appear at the top of the rank, from highest to lowest typicality scores, until 0.2 Cramer's value (threshold explained later). We denote *in italic* concepts identified by both SECA and ACE.

5.2.1 Sanity checks. Table 2 provides an overview of the interpretations generated by SECA for the bias injection datasets D1-D4. The results show that SECA identifies all those biases we injected. For instance, for D1, concepts around hour and minute are correctly picked up by the statistical tests, the mined rules and the decision tree and associated to the `female` class, while the NOT operator provides the concepts corresponding to their absence in the `male` class. The AND operator and the pairs of types and attributes identify the correct combinations of concepts also in the colored date and date city cases. The output include few possibly irrelevant concepts, always having Cramer's value below 0.2. These concepts are either outliers, i.e. concepts that impact the model's behavior at a low frequency, or noise from the saliency maps (concepts that are spatially close to the main salient visual elements). For instance, the concept `coat` (not in the table, Cramer's value 0.19) is significant in D3, as it always appears next to the text elements, and it is present in 13% and 2% of the `female` and `male` images respectively.

5.2.2 Concept correctness. SECA also provides relevant concepts for the learning set-ups with biases induced by resampling (D5, BM1.2, BM2.2), as shown in Tables 3 and 4. For instance, for BM1.2, concepts matching the background bias are uncovered, e.g. `water` for the `shark`, `grass` and `trees` for the `tench`, and `plate` for the `lobsters`, while these concepts are not identified as relevant in BM1.1. For D5, identified concepts match with the orientation bias such as hair-related concepts for `females`, and face-related concepts for `males` (e.g. `cheek`, `jaw`, `nose`), while for the "unbiased" task, the concepts focus on the hairstyle. The NOT operator exposes even more the bias, since concepts that combine the hair and NOT an element of the face appear more typical than only the hair (e.g. `hair AND NOT nose`). When comparing the two machine learning models M1, M2, 7 out of the top 10 concepts are the same but with a different ranking, reflecting that the models learned similarly but still with differences. For example, the `shark fins` and `tench heads` are used by Inception V3 and not VGG, which instead looked at the presence of a `shark head` with a higher typicality score.

The typicality scores are also relevant, as they are similar for concepts that appear with comparable frequency in the different classes. The scores evolve correctly when comparing models' behaviors: e.g., simple hair concepts have around 0.7 Cramer's value in the orientation bias data (D5.2) but are not even significant for the "unbiased" case (D5.1) since the model needs hair length.

5.2.3 Concept Coverage. Compared to ACE as shown in Tables 3 and 4, SECA generally provides a more complete set of correct concepts, allowing for a more accurate understanding of a model's behavior. ACE identifies mainly concepts that models rely on to classify images from every class, thus not discriminative (e.g. `wheel` is used to identify both ambulances and vans); these are also identified by our frequency analysis. SECA also uncovers certain entity

Table 3: Interpretations outputted by SECA using statistical testing and by ACE on the different learning task set-ups. Concepts in *italic* are captured both by SECA and ACE.

Bias	Met.	Interpretations (rank - Cramer's or TCAV value)
Fish (T2)		
yes	SECA	tench_body(1-.9), lobster_claw(2-.83), blue-water, green, *beige*, water(6-.7), face AND tench_body(8-.67), face(10-.65), grass(14-.58), green-grass(14-.58), trees(19-.47), plate(25-.35)
	ACE	*white* OR *light-grey*(1-.99), *white* OR *beige*(2-.9)
no	SECA	lobster_claw(1-.9), tench_body(2-.86), shark_body(3-.82), *grey-shark_body*(4-.81), *orange*(5-.8), *orange-lobster_claw*(6-.79), shark_fin(7-.69), tench_fin(9-.67), water, *water* AND *shark_body*(12-.6), *yellow-green*(14-.57), *white-plate*(32-.31)
	ACE	*orange-lobster, grey-blue water* OR *shark_body, grey-shark, blue- water* OR *blue-shark_body* OR *grey-shark_body*, blue OR grey OR green back, *yellow* OR *grey*(1-1.0), grey shirt OR tench(2-.96), *white-dish*(3-.86)
Vehicle (T3)		
yes	SECA	light(1-.61), blue-light(3-.53), orange blue(4-.46), blue-light AND grey-car_side(5-.45), stripe-car_side AND orange-car_front(6-.43), cross, light AND cross(9-.39), *road*(10-.32), *chassis* AND *wheel, black-car under*(11-.28)
	ACE	*light_grey-car_side* OR *sky* OR *road, black-wheel* OR *back, grey-road* OR *car_side* OR car_inside(1-1), *letters, black-chassis*(2-.98), *dark-grey* OR *black-wheel*(3-.97), *white-back*(4-.91)
no	SECA	stripe(1-.5), windowAND stripe(2-.5), stripe AND car_side(3-.46), stripe AND mirror(4-.44), stripe AND tire(4-.44), *orange, orange-stripe*(5-.38), stripeAND chassis(6-.28), white(15-.2)
	ACE	*black-bumper, black-tire* OR *gray-tire, black, orange* OR *red*(1-1.0), *gray-window* OR *gray-bumper*(2-.99), *black-chassis*(3-.69), *black* OR *gray*(4-.18), *tire*(5-.15), *white-sky*(6-.05), *orange-letters* OR *red-letters*(7-.01)

Table 4: Interpretations of SECA using statistical testing, rule mining and decision trees and of ACE on the gender classification task with and without orientation bias.

Cl.	Met.	Interpretations (ranges of typicality score)
Orientation bias (D5.2)		
F	Stat.	hair, black-hair(.7-.6), long, long-hair, black-hair AND long-hair(.6-.4), shirt AND hair, medium-hair(.4-.2)
	Rule	long AND grayAND black, long-hairANDblack-hair, long-hair, long(1.8-1.6), black-hair AND gray-back(1.4-1.1)
	Tree	long(.275), black, road, white, red(.06-.02)
	ACE	dark-gray hair OR shirt(1-.97), gray shirt OR back(.8-.6)
M	Stat.	neck(.7-.6), cheek, cheek AND neck(.6-.4), jaw, cheek AND jaw, face, neck AND jaw, nose, shirt AND cheek(.4-.2)
	Rule	hair AND neck, black AND short, black-hair AND short-hair, short (1.6-1.4), neck, hair AND ear, ear(1.4-1.1)
	Tree	car, neck, forehead, short, ear(.06-.02)
	ACE	gray, white OR gray shirt, gray sidewalk OR shirt(1-.97), light-brown skin(0.8-.6)
No injected bias (D5.1)		
F	Stat.	long, long-hair, longANDblack, long-hair AND black-hair(.6-.4), long-hair AND gray-back, gray-sidewalk-hair(.4-.2)
	ACE	gray-sidewalk, gray-back, brown-hair OR back(1-.97)
M	Stat.	short, short-hair, black-hair AND short-hair(.6-.4), short AND gray, neck, hair AND neck, short AND brown, ear(.4-.2)
	ACE	white-shirt OR back(1-.97), gray-sidewalk(.8-.6)

types present in single classes, that are missed by ACE (sometimes ACE outputs some color attributes that might relate to them). For instance, in D5.1, ACE outputs mostly colors that appear possibly in pair with entity types, e.g. brown color from hair or background for the female class, white color with a shirt or background for the male class, gray color for both classes. Our frequency analysis showed that these colors are salient in both classes rather equivalently (e.g. gray appears in 59% of female and 68% of the male images, gray background in 22% and 30% respectively), meaning they are not the solely used concepts. ACE does not provide any additional insights, but SECA also uncovers concepts relevant for individual classes, primarily related to hair length and presence of ear and neck for the male class –entities often hidden under the hair in the female images.

5.3 Results: Informativeness

The results obtained on the "unbiased" set-ups (BM1.1, BM2.1, D5.1) in Tables 3, 4 show that we not only obtain correct concepts, but these concepts are also highly informative about a model's behavior, whereas concepts identified by ACE provide fewer and less actionable insights - the prevalence of color-related concepts over entity type-related concepts makes, arguably, dataset modification more difficult. Particularly, the interpretations provided by SECA are more clear and intelligible, more diverse, and more precise.

5.3.1 Concept Intelligibility. ACE mainly highlights color related concepts that we can only sometimes associate with entity type concepts. In contrast, our approach outputs more fine-grain concepts with diverse entity types. This is probably due to technical limitations of the clustering algorithm used in ACE, that cannot precisely cluster entity types, but mostly color attributes. For instance, in BM1.2, ACE highlights white, light gray (probably coming from the plate, or from face or hand color), the gray color (shark skin or the background) for the shark, etc. These concepts are probably all correct, but are difficult to interpret since their provenance is not certain. Our approach on the contrary identifies the entity types that these attributes are associated to (e.g. green-grass, blue-water), thanks to the entity type-attribute pairs. Similarly, in D5.2 Table 4, ACE associates the female label to dark (hair or background) and pale colors (clothe or background), and male to pale and gray colors (clothe, background or faces). While it seems incorrect compared to our approach, extrapolating with our knowledge of the task, we see that they partly relate to face or hair concepts (i.e. the injected biases). Consequently, our interpretations are more actionable as concepts are traceable to visual entities in the dataset. Identifying pairs of entity types and attributes allows to uncover surprising and spurious biases, that are not clearly exhibited by ACE, but on which the dataset could be redistributed to mitigate the biases. For instance, in D5.1, SECA shows that the model primarily relies on the hairstyle, especially the stereotype of long / short hair, rather than pedestrian morphology. It also exhibits strong correlations between hair and dark colors, due to the low diversity of the dataset collected solely in Hong Kong.

5.3.2 Concept diversity. The diversity in the nature of the concepts outputted by SECA, such as concept combinations and absence of concepts, allows to uncover richer behaviors than with ACE in Table 3. For instance, in BM2.1, SECA shows that a) the co-occurrence of a vehicle side view and a colored stripe indicates an ambulance, but the co-occurrence of this view and a chassis indicates a van according to the statistical tests; b) the co-occurrence of a white vehicle side, a black tire and an orange stripe indicates an ambulance according to the mined rules; c) not having stripes and flashing light or having stripes and no light are associated with the van respectively with 0.47 and 0.44 Cramer's value (stripes are often indicative of ambulances), using AND and NOT operators. ACE misses these correlations that require the identification of absence concepts and the ability to calculate the significance of multiple concepts simultaneously – it would require image patches with multiple concepts represented next to each other, like a tire and a flashing light.

5.3.3 Interpretation Richness. The exploration tools of SECA allow to explore various, precise model behaviors that other approaches do not uncover, and that might not be straightforward to query.

While the frequency-based analysis and the statistical tests identify simpler significant concepts (in validation, they allow the user to query combinations of concepts however), association rule mining uncovers more complex combinations, e.g. Table 4 "long AND gray AND black" has the highest typicality. Simply by varying the configuration of the rule mining algorithm, it is possible to focus on diverse interpretation goals, such as finding frequent concepts by filtering out concepts with low support, or finding complex concepts that are less frequent by lowering such threshold. E.g., in BM1.1, the rule tench head AND tench body AND tench fin has a top lift score but is fairly rare in the data, hence it is outputted only with a support threshold under 0.2.

Decision trees discover complex behavior rules, and the information attached to them tell how common they are. For instance, in D5.1, the tree shows that NOT long AND NOT ear AND NOT background AND NOT black AND NOT road classifies males with 96% accuracy for 25 out of 300 records – which matches the intuitions about the data obtained from the statistical tests. Concepts appearing in the higher parts of the trees are accurately distinctive of the two classes (e.g. long hair is the first identified concept). Concepts in lower level do not correspond to expectations for the unbiased tasks: background elements appear as salient as parts of the body such as the ear or neck that we found are more important using the other methods. Because there are many visual elements but few rows in our tabular data (e.g. 78 elements for the pedestrian scenario and 300 records), the tree overfits to the data – curse of dimensionality– as confirmed by the low importance scores. Hence, only rules with high accuracy should be extracted from the branches, accounting for their frequency, and only the first levels of the tree should be used to extract individual concepts when few data are available.

5.4 Discussion

Results show that SECA correctly identifies different types of biases in model behavior –biases of visual entities, those arising from skewed data distribution and those from model architecture– and that it generates a rich set of interpretations for exploratory analysis of model behavior. Compared to ACE, SECA identifies a larger and more diverse set of concepts that are useful to identify more (biased) behavior patterns of a model. In particular, SECA identifies concepts with entity types and those comprising multiple sub-concepts that are often missed by ACE. We also observe that the different analysis tools of SECA allow to uncover various model behaviors.

A clear experimental limitation is the lack of an exact ground truth for what a model learns, making it challenging to conduct a full evaluation (especially in terms of interpretation completeness). We cope with this issue by setting up controlled experiments with manually induced biases of various types, which allow to evaluate interpretation effectiveness and informativeness from the bias angle. Another area of improvement concerns the amount and diversity of learning tasks and datasets. However, we stress that to date ours is one of the most comprehensive interpretability evaluation effort.

6 COST PERFORMANCE TRADE-OFF

In this section, we investigate *Q3: how do the main parameters that configure* SECA *impact the trade-offs between cost, correctness, and informativeness of the interpretations?*

6.1 Experimental set-up

6.1.1 Evaluation process. We study the impact that number of annotated images, annotation granularity, and the type of annotators (i.e., crowd-workers vs. trained annotators) have on the correctness and informativeness of the explanations generated by SECA. We use the same tasks as in the previous section.

Number of annotated images. As a reference, we use SECA to create interpretations based on a high number of annotated images (400). As we have shown in the previous section, SECA can generate satisfactory quality interpretations, i.e. interpretations that match the reference ones. We incrementally create interpretations from lower numbers of annotated images (between 20 and 400, in increments of 10). Finally, we compute the precision and recall of the concepts and the mean absolute error of Cramer's values, comparing the interpretations using smaller labeled image sets to the reference with 400 labeled images.

We hypothesize that the complexity of a learning task, which depends on a dataset characteristics, impacts the number of images needed to obtain similar correctness. The more classes to learn (need to uncover behaviors for more classes), the more diverse the visual entities and attributes per class (forces the model to use more concepts for classification), and the more concepts co-occur across classes (a model might rely on complex combinations of concepts), the more images should be needed to uncover a model's behavior. We investigate this by comparing the metrics computed on biased and unbiased scenarios (variation of intra-class semantic content diversity), and across tasks (more classes and lower inter-class concept co-occurrence in the fish task T2 than in T1 and T3).

Annotation granularity. We vary granularity from large to fine grained for both entity types and the attributes, defining different categories: for the entity type granularity category *E1*, all visual entities inherently part of the class (e.g. a blue star for the ambulance class, an antenna for the lobster class) are annotated as the class name, and all background objects are annotated as "background". In category *E2*, we distinguish the different parts of classes (e.g. claw, antennas, legs, body, head for the lobster), and we categorize background elements into large-grain categories (e.g. nature, food). Finally, in category *E3*, we refine the background annotations (e.g. rice, tomato, pavement) and the non-background ones when finer-grain entities can be identified. For the attributes, the category *A1a* combines color variations into seven main colors, and textures into large categories; in category *A1b* colors are combined depending on dark or light aspects. In category *A2* no combination is performed. We consider the reference granularity being the finest-grain ones, i.e. *E3* and *A2* and compare the resulting interpretations with coarser granularity categories.

Annotators. We compare the interpretations originating from saliency maps annotated by trained annotators (the authors) with saliency maps annotated by crowd workers, also computing the precision, recall, and mean absolute error.

6.1.2 Evaluation details. Experiments on number of annotated images. For the three learning tasks, we annotate 800 images, sample 400 images to form the reference interpretations, and sets of k images among the 400 remaining ones to form the interpretations to compare. We repeat this process 10 times to obtain statistically significant measures. We hypothesize that the precision and recall will be low for concepts with low Cramer's value. To verify this, we divide the reference concepts into 5 batches with Cramer's values equally divided between 0 and 1 (i.e., between 0 and 0.2, 0.2 and 0.4, etc.), and compute the recall per batch with all the concepts to compare with. We cannot do this for the precision as we cannot directly compare the reference batches to comparison concepts – small errors in Cramer's values would make the measures wrong (e.g. a comparison concept of Cramer's value 0.61 would lower precision if its reference concept is in the batch $0.4 - 0.6$). Instead, we simply count the number of wrongly retrieved concepts in the comparison set. We also compute the mean absolute error per batch

as we hypothesize that low Cramer's value concepts are attributed less accurate values due to the sampling error.

Experiments on annotators. In this experiment, we compare annotations of trained annotators to untrained crowd workers recruited on crowdsourcing platforms, focusing on general annotation properties (like amount of bounding boxes, coverage of the salient areas, amount of time spent, feedback questionnaires, etc.), and we investigate how the provided concepts compare semantically. For this semantic comparison, we automatically map concepts provided by crowdworkers to those provided by the trained annotators by computing a similarity score between the concepts word embeddings, fixing a threshold T and retaining as matching only the concepts with similarity above T. We repeat this with the different annotation granularities. Assuming that the authors' annotations are indeed of high quality, we can now investigate the precision and recall of the crowd compared to the authors. Furthermore, we investigate the effect of annotation reconciliation (step C3 of our approach) which is necessary when multiple crowdworkers provide annotations with varying vocabulary.

Crowdsourcing component implementation. We deployed the annotation task on Amazon Mechanical Turk. Each HIT was composed of a set of ten images and their saliency maps, and was assigned to three crowd workers[7]. The instructions encourage the workers to search for domain knowledge to give precise annotations as a pilot study showed diverse annotation precision. $k = 125$ clusters are used for the reconciliation component as it provides the best Silhouette scores.

6.2 Results: number of images

Figure 4 shows an example of the curves obtained for the fish task BM1.1 (results for other datasets are similar, and reported in the companion page). We observe that 300 annotations provide satisfactory concept sets and Cramer's values, and only 200 annotations are needed if we do not need to identify less significant concepts. We do not observe significant differences across tasks and biases.

Recall. For all the learning tasks, concepts are retrieved with only 200-300 annotations. Although the overall recall might not seem satisfying even for 400 annotations, the recall for all concepts with Cramer's value greater than 0.2, closely approaches 1 (and 0 standard deviation) with 300 annotations, and a minimum of 0.9 recall is observed with 200 annotations. Concepts of Cramer's value between 0.4 and 1 are even retrieved with just 100 annotations. Lower Cramer's values are indicative of less significant, possibly irrelevant concepts (see subsubsection 5.2.1), picked up by a model in lower frequencies, thus they are more susceptible to sampling noise, and need more images to be retrieved.

Recall curves are similar across tasks. For instance, BM1.2 also needs 300 images but with a standard deviation lower than BM1.1, probably because of its lower intra-class complexity. D5.1 (pedestrian) just requires 20 more images to approximate a recall of 1 with a standard deviation lower than 0.02 –probably due to more inter-class co-occurrences than BM1.1. Generally, this is because the impacts of the data characteristics balance each other, e.g. although there are more classes in T2 (fish), the image content in T3 (vehicle) or T1 (pedestrian) is more diverse.

[7]We included workers from UK and USA with at least 5K approved hits, and a HIT approval rate greater than 85%.

Table 5: SECA interpretations on the fish bias task for various Granularity of entity types. Granularity *E3* is in Table 4.

Gra.	Interpretations (Cramer's value)
E1	lobster(.95), tench(.92), shark(.83), back AND lobster(.89), tench AND back(.88), orange(.81), grey-tench(.83), orange-lobster(.79), green-back(.78), light grey-back(.74)
E2	tench_body(.89), lobster_claw(.85), lobster_body(.73), orange-lobster_claw(.72), blue-water(.75), water(.7), beige-human_body_part(.63), food(.46), table_tool, clothe(.4)

Precision. The precision is also satisfying with only 200 images. The precision curve decreases from 1 with 10 images to 0.93 with 200 images and 0.9 with 300 images, the standard deviation remains constant at 0.04. A closer look at Figure 4b shows that, once more, most incorrect concepts have Cramer's values inferior to 0.2 when increasing the number of images since such concepts are more subject to sampling noise. Not accounting for these concepts allows to keep a precision higher than 0.9 for every number of images.

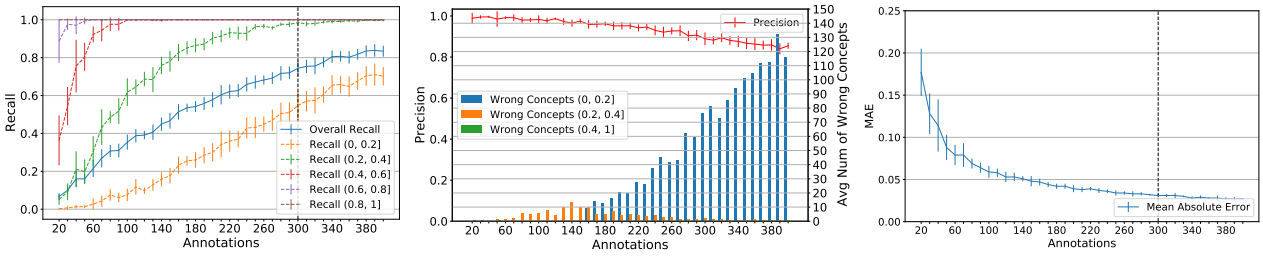
The curves are similar across tasks, with T1 and T2 having a larger standard deviation around 0.1 and 0.07 respectively, verifying our hypotheses. Only tasks with many more classes and higher visual entity intra-class diversity or inter-class co-occurence would probably require to annotate more images.

Mean absolute error. The approximation of Cramer's values is accurate even for less than 200 annotations (again except for concepts of Cramer's values below 0.2). The error decreases rapidly with more images, going from 0.2 with 0.1 standard deviation for 20 annotations, to 0.026 and 0.001 standard deviation for 300 images and above. This is because having more annotations allows to approach the real joint distribution of concepts and classes in the data, on which the Cramer's values are computed.

6.3 Results: granularity of the annotations

Entity types. We report the results on BM1.2 in Table 5 (results from other tasks point to similar conclusions, and are reported in the companion page). With large grain annotations (*E1* and *A1a*), the retrieved concepts are correct but poorly informative as actionable insights. E.g., `lobster`, `tench` and `shark` are the most salient concepts, followed by color concepts, combinations of the background concept and one of the previous fish-related concepts, or pairs of color and fish concepts (e.g. `orange-lobster`). This interestingly indicates that the model uses both concepts related to the classes and background concepts, but without more details we can neither conclude about the validity of this behavior – certain background concepts could make sense, e.g. shark in the water, nor identify visual background entities to redistribute in order to remedy to the potential background bias.

Finer grain annotations bring more precise debugging information. For instance, *E2* uncovers the different parts of the concept classes (e.g. `lobster claw`) possibly in combinations with colors (e.g. `orange-lobster claw`), and the background entities (e.g. `blue-water`, `beige-human body part`) used by the model and based on which a dataset can be transformed to mitigate biases.



(a) Recall. **(b) Precision and number of incorrect concepts.** **(c) Mean absolute error.**

Figure 4: Analysis of the number of image annotations required by SECA for the ImageNet Fish task. The values in brackets correspond to the Cramer's values of the reference concepts used to compute the corresponding curves.

Further detailing background concepts *E3* provides even more detailed information, e.g. the `face` is the human body part the most associated with the `tench`, the `shirt` is the most associated cloth.

Hence, depending on the interpretation need, the granularity of annotations needed differs. The medium granularity is enough to explore the general functioning and validity of a model, while the finest-granularity provides precise information to mitigate behavior biases. If the finest granularity is employed, we recommend to obtain a higher level overview of the model's behavior by querying combinations of concepts with an `OR` logic connection –equivalent to aggregating concepts into larger grain ones. For instance, the medium granularity uncovers shark-related concepts with Cramer's values around 0.65 or lower, while aggregated altogether the value increases to 0.83, above the background concepts (0.74), showing the potential correctness of the model's behavior.

Entity attributes. The granularity of attributes on the contrary do not lead to differences that impact the interpretations of the models. This is probably due to the limited range of distinct colors that a human is able to annotate easily. Automatic annotation methods using pixel values might bring additional insights on the color shades that are the most important for classification.

6.4 Results: Crowd vs. Trained Annotators

6.4.1 Components' quality. ***1)*** *Annotations.* Crowd workers took $\mu = 28m, \sigma = 11$ minutes to execute the task. Quality of annotation was good. Most workers who took less than 15 minutes provided 1-2 annotations of simple salient areas per images, while the ones who took more time provide 2.8 annotations per image in average, with a maximum of 66 per HIT. The difficulty of identifying salient areas, drawing bounding boxes and annotating entity types and attributes was evaluated with an average of 3.3, 3.1, 3.1 and 3.3 respectively on a scale of 1 (easy) to 5 (difficult). Few annotators provide full coverage of the salient areas, either due to not identifying certain entities, or due to not drawing boxes around the entire areas. This has limited impact on interpretation quality, as having precise bounding boxes is not important, and using multiple annotators proved to provide the needed coverage.

2) *Annotation Reconciliation.* The clustering approach used to determine reconciled annotations is satisfactory: most clusters are relevant for the interpretation task. They reconcile wording differences (e.g. `tooth` and `teeth`), synonyms and terms that designate similar concepts (e.g. `belly`, `stomach`). Mistakes are introduced by words with multiple meanings, e.g. lobster `antenna` is grouped with network infrastructure words, because no context is used to create the embeddings. Some terms that relate to different granularities are grouped (e.g. `hand`, `fingers` and `thumb`), which might impact the interpretations when the finest granularity is needed.

6.4.2 Correctness of the interpretations. We report here the results for the fish bias task. The interpretations from the crowd uncover the main expected biases, e.g. presence of `water` for the `shark` images, `grass`, `trees` and `human body parts` around the `tench` images, `plates` for the `lobster` images, and only a few concepts do not appear, e.g. certain food concepts such as `corn` for the lobster.

However, we obtain only 0.48 precision, 0.61 recall and 0.18 mean absolute errors of Cramer's value on significant concepts retrieved for the finest granularity. The medium and large granularity respectively reach a precision of 0.53, 0.57, a recall of 0.70, 1.0 and a mean absolute error of 0.19, 0.40. Only a few concepts are not mentioned by the crowd (e.g. `lemon`), probably because they appear small in the background of the images, behind the main objects. As hinted by these increasing values, this contradiction is mainly due to measurement errors: differences in the vocabulary and granularity of annotations cause errors in the mapping used in the evaluation process, which makes precision and recall low. Most reference concepts that appear as missing from the crowd interpretations are actually retrieved. For instance, the concepts from the trained annotators `shellfish` and `sauce` are annotated by the crowd with `oyster`, `shrimp` and `soup`, `liquid`. The crowd annotations are often more fine-grained, which also lowers the precision. For instance, heads annotated with `boy head`, `man head`, `woman head` and some with `human head` instead of solely the latter like the trained annotators', formed two distinct clusters (`human` associated with `animal` and the others together), one appearing irrelevant. The average mean absolute error increases with larger granularity because we modify only the granularity of the trained annotators' concepts, while the worker's concepts remain distinct with lower Cramer's values. Overall, employing the crowd with simple post-processing methods provides interpretations of similar correctness, with only few fine-grain concepts missing.

6.4.3 Informativeness of the interpretations. Certain interpretations obtained from the crowd are richer in terms of granularity than those from the trained annotators. For instance, the crowd interpretations differentiate between the shark `fins`, e.g. `caudal fin`, `dorsal fin`, whereas only `fin` appears in the trained annotators'

concepts. This is because certain workers provide precise vocabulary (as encouraged in the instructions) that a trained annotator might not have thought of (e.g. `pectoral`, `caudal`, `dorsal fins`, etc.), or for which a trained annotator does not have domain knowledge like the species of fish labeled by the crowd (e.g. `muskellunge`, `carp`, `tench`). This is the main advantage of using the crowd instead of trained annotators. Having multiple, lower cost, annotators allows to mitigate individual bias, as different persons focus on different entities, granularity and labels.

6.5 Discussion

SECA can produce correct and informative interpretations already with few images annotated (300) using crowd workers.

Significant concepts are well covered with even fewer images (100, Cramer's value above 0.4), with satisfactory performance. While finest-grain concepts are useful to understand precise model behavior and debug it, medium-grain concepts seem to be satisfying for model validation and general exploration purposes. Crowd annotations generally align with those from trained annotators, but with a richer vocabulary that allows to gain comprehensive understanding of model behavior. While workers' contributions are not always accurate, we stress the simplicity of our task design. Experiments show that crowd workers can be systematically employed to support saliency map annotations, thus enabling an accurate, scalable, and relatively cheap post-hoc interpretability method. We acknowledge though, that our experiment is limited to binary/three classes problems. Experiments on tasks with more classes can help quantify the impact of the class number and diversity on cost effectiveness trade-off.

7 CONCLUSION

We presented SECA, a framework to support post-hoc, interactive interpretation of machine learning models for image classification. SECA offers interpretations based on easily understandable semantic concepts (entities and attributes). These concepts are obtained via crowd-sourcing from local interpretatability saliency maps, and then reconciled and consolidated into a unified and structured representation which allows the use of different statistical mining techniques to discover or query for concepts relevant for a model's decision making. Extensive experiments showed that, compared to related work, SECA can discover more informative and complete concepts, and that these concepts are more interpretable and actionable to debug a model. Results show that using crowd workers to provide semantics to annotate salient image areas provides results with sufficient performance at lower costs, and that also smaller sample of annotated images lead to actionable results. As future work, we plan to investigate the mitigation of the spurious biases identified by our framework.

REFERENCES

[1] Alan C Acock and Gordon R Stavig. 1979. A measure of association for nonparametric statistics. *Social Forces* 57, 4 (1979), 1381–1386.
[2] Diederik Aerts. 2016. Quantum theory and human perception of the macro-world. *How Humans Recognize Objects: Segmentation, Categorization and Individual Identification* (2016), 210.
[3] D Aerts and L Gabora. 2005. A theory of concepts and their combinations I. *Kybernetes* (2005).
[4] R Agarwal and al. 1994. Fast algorithms for mining association rules. In *VLDB*.
[5] M Ancona, E Ceolini, C Öztireli, and M Gross. 2018. Towards better understanding of gradient-based attribution methods for Deep Neural Networks. In *ICLR*.
[6] S Bach, A Binder, and al. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* 10, 7 (2015).
[7] D Bahdanau, K Cho, and Y Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.
[8] A Barbu, D Mayo, and al. 2019. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *NeurIPS*. 9453–9463.
[9] L Breiman, J Friedman, and al. 1984. *Classification and regression trees*.
[10] J Deng, W Dong, R Socher, L-J Li, K Li, and L Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Proc. of 2009 CVPR Conference*. IEEE, 248–255.
[11] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608* (2017).
[12] Alex A Freitas. 2014. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter* 15, 1 (2014), 1–10.
[13] A Ghorbani and al. 2019. Towards automatic concept-based explanations. In *NeurIPS*.
[14] X Hu, H Wang, A Vegesana, and al. 2020. Crowdsourcing Detection of Sampling Biases in Image Datasets. In *Proc. of WWW*. 2955–2961.
[15] B Kim, M Wattenberg, J Gilmer, and al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors. In *ICML*.
[16] P W Koh and P Liang. 2017. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730* (2017).
[17] X Liu, H Zhao, M Tian, L Sheng, J Shao, S Yi, J Yan, and al. 2017. Hydraplus-net: Attentive deep features for pedestrian analysis. In *Proc. of IEEE ICCV*. 350–359.
[18] Nicolas Maleve. 2019. An Introduction to Image Datasets. (2019). https://unthinking.photography/articles/an-introduction-to-image-datasets
[19] Eric Margolis and Stephen Laurence. 2007. The ontology of concepts-abstract objects or mental representations? *Noûs* 41, 4 (2007), 561–593.
[20] B Nushi, E Kamar, and al. 2017. On human intellect and machine failures: Troubleshooting integrative machine learning systems. In *AAAI*.
[21] European Parliament and Council of EU. 2018. European Union General Data Protection Regulation. (2018). http://www.privacy-regulation.eu/en/13.htm
[22] Bing Ran and P R Duimering. 2010. Conceptual combination: Models, theories and controversies. *International Journal of Cognitive Linguistics* 1, 1 (2010), 65–90.
[23] V C Raykar, S Yu, and al. 2010. Learning from crowds. *JMLR* 11, Apr (2010).
[24] M T Ribeiro, S Singh, and C Guestrin. 2016. " Why should I trust you?" Explaining the predictions of any classifier. In *Proc. of the 22nd ACM SIGKDD*.
[25] A S Ross, M C Hughes, and F Doshi-V. 2017. Right for the right reasons: training differentiable models by constraining their explanations. In *IJCAI*. 2662–2670.
[26] O Russakovsky, J Deng, H Su, J Krause, and al. 2015. Imagenet large scale visual recognition challenge. *IJCV* 115, 3 (2015), 211–252.
[27] R R Selvaraju, M Cogswell, A Das, and al. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proc. of ICCV*. 618–626.
[28] A Shrikumar, P G, and A Kundaje. 2017. Learning Important Features Through Propagating Activation Differences. In *ICML*. 3145–3153.
[29] K Simonyan, A Vedaldi, and A Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *ICLR*.
[30] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[31] D Smilkov and al. 2017. SmoothGrad: removing noise by adding noise. (2017).
[32] E Štrumbelj and I Kononenko. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems* (2014).
[33] M Sundararajan and al. 2017. Axiomatic Attribution for Deep Networks. In *ICML*.
[34] C Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proc. of the IEEE CVPR*. 2818–2826.
[35] Global Legal Research Directorate The Law Library of Congress. 2019. Regulation of Artificial Intelligence in Selected Jurisdictions. *https://www.loc.gov/law/help/artificial-intelligence/index.php* (2019).
[36] Jennifer Wortman Vaughan. 2018. Making better use of the crowd: How crowdsourcing can advance machine learning research. *JMLR* 18, 193 (2018), 1–46.
[37] K Xu, J Ba, R Kiros, K Cho, A Courville, and al. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*. 2048–2057.
[38] Y Yan, G M Fung, and al. 2011. Active learning from crowds. In *ICML*. 1161–1168.
[39] C Yang, A Rangarajan, and S Ranka. 2018. Global model interpretation via recursive partitioning. In *IEEE HPCC/SmartCity/DSS*. IEEE, 1563–1570.
[40] J Yang, T Drake, A Damianou, and Y Maarek. 2018. Leveraging crowdsourcing data for deep active learning. An application: learning intents in Alexa. In *WWW*.
[41] J Yang, A Smirnova, and al. 2019. Scalpel-cd: leveraging crowdsourcing and deep probabilistic modeling for debugging noisy training data. In *WWW*. 2158–2168.
[42] M Yang and B Kim. 2019. Benchmarking Attribution Methods with Relative Feature Importance. (2019).
[43] M D. Zeiler and R Fergus. 2014. Visualizing and Understanding Convolutional Networks. In *ECCV*. 818–833.
[44] Q Zhang and al. 2018. Interpretable convolutional neural networks. In *CVPR*.
[45] Z Zhang, J Singh, U Gadiraju, and A Anand. 2019. Dissonance between human and machine understanding. *Proc. of the ACM on HCI* 3, CSCW (2019), 1–23.
[46] Minhaz Fahim Zibran. 2007. Chi-squared test of independence. (2007).