Oracle Issues in Machine Learning and Where to Find Them

Liem, Cynthia C. S.; Panichella, Annibale

DOI
10.1145/3387940.3391490

Publication date
2020

Document Version
Final published version

Published in
Proceedings of the 8th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE)

Oracle Issues in Machine Learning and Where to Find Them

Cynthia C. S. Liem
c.c.s.liem@tudelft.nl
Delft University of Technology
The Netherlands

Annibale Panichella
a.panichella@tudelft.nl
Delft University of Technology
The Netherlands

ABSTRACT

The rise in popularity of machine learning (ML), and deep learning in particular, has both led to optimism about achievements of artificial intelligence, as well as concerns about possible weaknesses and vulnerabilities of ML pipelines. Within the software engineering community, this has led to a considerable body of work on ML testing techniques, including white- and black-box testing for ML models. This means the oracle problem needs to be addressed. For supervised ML applications, oracle information is indeed available in the form of dataset 'ground truth', that encodes input data with corresponding desired output labels. However, while ground truth forms a gold standard, there still is no guarantee it is truly correct. Indeed, syntactic, semantic, and conceptual framing issues in the oracle may negatively affect the ML system's integrity. While syntactic issues may automatically be verified and corrected, the higher-level issues traditionally need human judgment and manual analysis. In this paper, we employ two heuristics based on information entropy and semantic analysis on well-known computer vision models and benchmark data from ImageNet. The heuristics are used to semi-automatically uncover potential higher-level issues in (i) the label taxonomy used to define the ground truth oracle (labels), and (ii) data encoding and representation. In doing this, beyond existing ML testing efforts, we illustrate the need for software engineering strategies that especially target and assess the oracle.

ACM Reference Format:
Cynthia C. S. Liem and Annibale Panichella. 2020. Oracle Issues in Machine Learning and Where to Find Them. In *IEEE/ACM 42nd International Conference on Software Engineering Workshops (ICSEW'20), May 23–29, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3387940.3391490

1 INTRODUCTION

In traditional software systems, oracles (e.g., test assertions) are used to specify what the expected output for a given input should be. The problem of determining whether a test passes or fails is referred to as the *oracle problem*, which is a well-known challenge in the software engineering community and has been widely discussed and investigated in the related literature [1]. These days, more and more software systems include machine learning (ML) components, or implement full ML decision pipelines. In such cases, new challenges arise with regard to the oracle problem [26, 28].

ICSEW'20, May 23–29, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7963-2/20/05.
https://doi.org/10.1145/3387940.3391490

Many present-day ML systems are based on supervised techniques, in which the learning of relevant patterns is based on datasets with many examples of inputs and desired corresponding outputs. Commonly, such datasets are partitioned into at least a training and test set. Together with this, the practitioner will specify the desired ML model (e.g., a deep neural network with a pre-specified architecture, along with the desired optimization criterion). From a software engineering perspective, this model pre-specifies unweighted decision-making logic. During training, based on the data in the training set, the weights of the model will iteratively be optimized. As a consequence, after successful training, a program should have been learned that, based on a given input, will yield the desired output as well as possible.

With scalable decision-making being the ultimate goal in many ML scenarios, a successful ML model should be able to generalize, i.e., making correct decisions for unseen input that is sufficiently similar to its training data. This is why it is common practice to not train on all available data, but to at least hold out a test set (and in some cases, a validation set, to be used during intermediate development). As the partitions originate from the same dataset, data points within them can be considered to be sufficiently similar. As the training procedure did not observe the samples in the validation and test sets, they can be considered as 'unseen' data to the trained model, but we do know upfront what the corresponding desired output should be.

In conclusion, supervised ML systems treat the oracle problem in a different way than traditional software systems. In traditional software systems, testing is used to assess whether the system behaves as intended. At that testing moment, the developer will need to establish the oracle. In case a test-driven development paradigm [3] is followed, this will be before the software is written, but the oracle alternatively may only be defined after the software is written. Furthermore, the source of the oracle can be completely independent of the creator of the software.

In contrast, in supervised ML systems, the existence of oracle data—in the form of training and test data—is a hard requirement for the system to be able to be built, and the realization of the software will be intimately tied to what this data reflects. Thus, while a supervised ML system is in the progress of being trained and evaluated, one could argue there is no oracle problem, and we know exactly what the system should output, according to the 'ground truth' as specified in the dataset. The dataset thus defines the gold standard against which the ML system will be assessed.

Of course, this does not guarantee that generalization will work as intended. Indeed, on unseen data, even powerful deep learning based ML systems have been observed to make mistakes that humans would not make, e.g., because of sensitivity to so-called adversarial examples [9, 16]. This problem has been acknowledged in both the ML and software engineering communities. From a

software engineering perspective, in these situations, the oracle problem re-emerges, and may provide even harder challenges on 'what should be correct' than in classical software systems [28]. To address this problem, in a similar fashion as in classical software systems, the software engineering community has proposed testing techniques for ML systems. Ben Braiek and Khomh [4], as well as Zhang et al. [28], provide extended overviews of such ML testing approaches in the recent literature. For example, research effort has been devoted to defining and assessing testing methods to reach high code coverage for ML models (e.g., [25]), increasing the quality of test data with mutation testing (e.g., [13]), testing the implementation of ML pipelines (e.g., [18]), and revealing defects using metamorphic relations (e.g., [15]).

Returning to the gold standard datasets on which ML systems will initially be trained and evaluated, by design, a perfectly trained ML system will perfectly match this gold standard. However, the gold standard is not actually guaranteed to be perfect. At a syntactic level, data values may be inconsistently or invalidly encoded, and in need of cleaning, verification, and sanity checking before they realistically can be used in ML pipelines; multiple proposals have emerged to (semi-)automatically address this problem [2, 11, 12, 21].

Beyond this, at a semantic and more conceptual level, human judgment of ground truth may have been partially erroneous, potentially inconsistent, or even ethically undesired to become part of an automated decision pipeline, e.g., because it may encode biased or offensive assessments [7, 23, 24, 27]. Such issues raise questions on the true quality of the oracle, will have a serious influence on what an ML system will infer, and, from an experimental validity perspective, may cause systems to not actually learn what they are supposed to learn [24]. Recognizing that these issues indeed are present—and conceptually problematic to a given ML use case—requires yet another step of human judgment. Therefore, they typically are only identified after qualitative manual analysis. Considering the typical scale of real-life datasets, it is, however, unrealistic that pure qualitative analysis will be able to yield truly comprehensive and systematic findings.

In this work, we will focus on ImageNet [8], a classical large-scale dataset that has played a key role in advancing ML success in the computer vision domain. More specifically, we focus on the ImageNet data partition used in the 2012 ImageNet Large-Scale Visual Recognition Challenge, which has grown into a standard reference benchmark for new computer vision ML models. Proposing two heuristics, based on information entropy and semantic analysis, and examining the outputs of multiple classical ML vision models, we surface various potential oracle issues. As we will show, these issues occur with regard to label taxonomy and information representation. As we will argue, for certain ML use cases, these could be explicitly harmful to a practical system's integrity.

2 THE IMAGENET USE CASE

ImageNet is a well-known, large-scale hierarchical image database [8]. It has partially been inspired by WordNet, a well-established lexical database of English words grouped in sets of synonyms, called *synsets* [14]. Seeking to provide visual illustrations of WordNet synsets, ImageNet provides a large-scale visual ontology to the computer vision community. For this, millions of images were collected from the Internet and labeled through crowdsourcing.

As ImageNet developed, multiple academic benchmark initiatives were run on its content [20]. Out of these, the 2012 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC2012) has been of particular significance, as this was the year in which deep learning techniques—more specifically, large-scale convolutional neural networks—started becoming successful. Ever since, deep learning techniques have become omnipresent in computer vision, and the ILSVRC2012 classification challenge has become a standard reference for reporting ML performance.

The ILSVRC2012 classification challenge focused on classifying images into one out of a thousand object categories. Formally framed, image input data $\mathbf{x}$ should be mapped to one of the image class IDs $\{0, \ldots, 999\}$, corresponding to the 1000 categories of interest. For each image in the dataset, a single ground truth label is available. However, in practice, many deep networks will output a distribution vector $\hat{\mathbf{y}} = [\hat{y}_0 \cdots \hat{y}_{999}]$, with $\hat{y}_i = P(y = i|\mathbf{x})$, i.e., the probability of each of the class IDs being evidenced, given input $\mathbf{x}$.

The ILSVRC2012 challenge provided a training set of more than 1.2 million images, a validation set of 50K images for use during development, and a test set of 100K images, for which ground truth labels were released at the end of the challenge. As for the reference status of ILSVRC2012, commonly reported 'results on ImageNet' in ML literature typically refer to the results obtained on the ILSVRC2012 object classification challenge. The authors of many successful deep architectures offer pre-trained weights that were trained on the ILSVRC2012 training set, and report reference performance on the ILSVRC2012 validation set. This performance is typically expressed through an accuracy measure, expressing the fraction of images for which the ground truth class is identified as the highest-probability class (top-1 accuracy), or among the five highest-probability classes (top-5 accuracy). The latter measure has been suggested as a more robust measure, capable of dealing with the phenomenon that many images may depict more than one object, even though each image only bears a single class label [20].

3 SURFACING POTENTIAL ORACLE ISSUES

In this section, we describe how we will surface potential oracle issues in the ground truth, considering two heuristics and a comparison technique inspired by differential testing.

3.1 Alternative models as alternative experts

Differential testing is a well-established technique aimed at comparing the output of alternative implementations against the same input. Disagreements in the test execution output across the alternatives expose potential defects in the implementations. For example, in the context of ML implementations, CRADLE [18] uses differential testing to find bugs in deep learning libraries through cross-backend validation.

In this work, we apply a strategy inspired by differential testing to surface potential dataset oracle issues. Multiple alternative ML models can be trained on the same data; as they were optimized through a data-driven programming paradigm [28], a model that is reported to perform well can be assumed to adequately reflect the main patterns in the data it was trained on. Therefore, if multiple

alternative models with strong reported performance disagree with the ground truth in consistent ways, rather than signifying this is a problem in either of the models, this may signify there is a problem with the ground truth.

We re-emphasize that this strategy targets a different aspect of an ML pipeline than traditional differential testing. Traditional differential testing, as e.g. applied in [18], focuses on testing alternative implementations of the same model. Therefore, for such methods, the data and model needs to be fixed, while variations will be examined in backend libraries. In our case, we will fix the data and the backend libraries, but vary the models. With model performance being reported as an overall accuracy number, two models with identical performance are not guaranteed to have made identical judgments. That is, two models with a top-5 accuracy of 0.9 will both have recognized the correct category for 90% of the tested data instances, but it is not guaranteed that the 90% that each model got correct considers the exact same sample. Therefore, metaphorically, each strong-performing model can be seen as an expert on the data, but different experts may be focusing on different salient aspects of the same data. Hence, if multiple experts disagree on the same input, they may be pinpointing alternative salient aspects of it. Yet, if they disagree in consistent ways, other aspects may be salient than the ones that are formally encoded.

Considering our current dataset of focus, the ILSVRC2012 data in ImageNet, the traditional top-1 and top-5 accuracy performance metrics only care about the presence of the ground truth class in the top-1 or top-5 of a model prediction. However, as discussed in the previous section, typical present-day deep learning models actually give a richer output than that: for each image input, they will yield a probability distribution vector, indicating the model's confidence for all possible object classes that can be predicted for. We use this to deepen our understanding of the judgments of our different models: a model that has 0.99 confidence on a single class is 'more sure' of observing that particular class than a model yielding more uniformly distributed probabilities for multiple classes at once.

This notion can be numerically expressed through the *Shannon entropy*, which is a traditional measure of information content:

$$H(\hat{\mathbf{y}}) = -\sum_{i} P(y = i|\mathbf{x}) \log_2(P(y = i|\mathbf{x})) \quad (1)$$

Smaller values of $H(\hat{\mathbf{y}})$ indicate there is less information content in $\hat{\mathbf{y}}$; in other words, the probabilities will be concentrated over fewer object classes, and model predictions can thus be assumed to be confident and predictable. In contrast, larger $H(\hat{\mathbf{y}})$ indicate higher information content in $\hat{\mathbf{y}}$: probabilities will be distributed over more object classes, and outcomes thus are less clear-cut. By computing and analyzing $H(\hat{\mathbf{y}})$ for predictions made by different models, we have a heuristic that can help us uncovering possible oracle issues, in particular in relation to ambiguity and observability of object classes.

3.2 Semantic Analysis

When creating the ground truth, design decisions should be made on what classes and labels should be represented. For certain problems, the labels are very straightforward. For example, in defect prediction, a class (or method) is either `defective` or `non-defective`. These two labels are mutually exclusive. However, for other problems, choosing the appropriate labels is more complicated. For example, in image recognition, an image can contain multiple objects. Furthermore, the same objects can have alternative labels (e.g., synonyms). In this case, engineers may choose to use only one label or to allow using multiple labels for the same picture, e.g., for the different objects in the image or considering possible synonyms.

This design choice will affect how the oracle information will be represented in the ML pipeline. Generally, the information in an ML pipeline takes numerical forms, and will be processed by mathematical frameworks. When considering ML classification tasks, typically, the different possible object classes will be implicitly modeled as being independent and equivalent. However, in the real world, there may be different relations between object classes. Considering ImageNet, the original purpose of the ImageNet dataset indeed was to offer a hierarchical visual ontology; furthermore, the WordNet knowledge source that drove the ImageNet development explicitly encodes relations between the entities inside of it.

Depending on the ultimate ML use case, being explicitly aware of these relationships may be of importance. When only considering the traditional ILSVRC2012 focus on 'whether an object of class i is present in an image', this may be less relevant. However, when moving towards broader visual understanding and reassessments of the trustability of existing oracles, certain classification 'mistakes' may be more logical and explainable in comparison to the ground truth than others.

With WordNet including synsets and connections between them, according to various semantic relations, we can use it to uncover possible ambiguities in our data labels. For the sake of our analysis, we use the 1000 ILSVRC2012 class label names as queries to the WordNet database, and then search for pairs of class labels characterized by three semantic relations: (1) synonyms, (2) homonyms, and (3) meronyms. In our context, *synonyms* are pairs of class labels that are treated separately in IlSVRC2012, but that have the same meaning. Synonyms can lead to ambiguities in data labels, as the same object in an image might be labeled differently by different human validators. *Homonyms* are pairs of class labels that are spelled and pronounced the same, but that have different meanings. The labels of homonyms should be appropriately stored when building the datasets to avoid future misinterpretations. Finally, *meronyms* are pairs of labels linked by a 'part of' relation. Meronyms may lead to ambiguities, since an image that contains one object/label might also contain objects relating to the other paired label.

4 ORACLE ISSUES IN IMAGENET

4.1 Approach

Using the techniques discussed in the previous section, we focus on semi-automatically surfacing oracle issues.

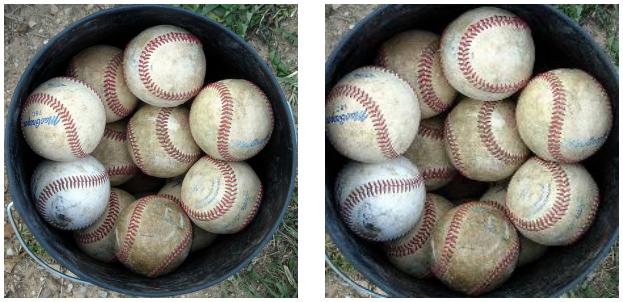
In choosing our 'expert' models, we choose four well-known deep vision models: `vgg16` [22], `vgg19` [22], `ResNet50` [10] and `ResNet101` [10]. For all these models, we use the pre-trained weights as released by the original authors after training on the ILSVRC2012 training set, and as released through Keras [6]. More specifically, we run the models using Anaconda Python 3.5.6, Keras 2.2.2, and a Tensorflow 1.10.0 backend. All models require input data to be

Relationship	#Count	Pairs
Homonyms	6	<crane (bird), crane (machine)>, <cardigan (jacket), Cardigan (dog)>, <polecat (skunk), polecat (fitch)>, <sunglass (convex lens), sunglasses (shades)>, <maillot (tank suit), maillot (dance suit)>, <gong (tam-tam), gong (bell)>
Synonyms	11	<loggerhead turtle, Caretta>, <cock, hammer>, <fly, wing>, <coil, ringlet>, <tub, bathtub>, <projectile, missile>, <screen, shield>, <harp, harmonica>, <drum, barrel>, <laptop, notebook>, <corn, acorn>
Meronyms	2	<palace, castle>, <castle, stone wall>

Table 1: Semantic relationships between labels in the ImageNet ILSVRC2012 dataset.

(a) Original **(b) Cropped**

Figure 1: Image ILSVRC2012_val_00017853, class kneepad, original vs. cropped version. Borders drawn for clarity.



(a) Original **(b) Cropped**

Figure 2: Image ILSVRC2012_val_00006594, class bucket.

$224 \times 224 \times 3$ pixels in dimension. For this, we follow the cropping procedure from [19][1], which was shown to reproduce the validation performance results reported in Keras. On the ILSVRC2012 validation data, each of the models has a top-5 accuracy > 0.9.

We run the 4 models on all 50K data instances in the ILSVRC2012 validation set, and compute and analyze the Shannon entropy $H(\hat{\mathbf{y}})$ for each of the predictions. Following the concept of understanding how our models may 'agree to disagree' with the official ground truth, we focus our analysis on situations in which *none of the four deep vision models manage recognizing a ground truth class amidst the top-5 predictions.* This holds for 2269/50K images (4.5%) in the validation set.

To understand whether these mistakes are true mistakes, or may be explained by oracle issues, we consider what types of classification mistakes are being made, and if they may be explainable, considering potential semantic relations between different object classes. For this, we use the results of our semantic analysis of the 1000 object class categories in WordNet. A full summary of found relations is given in Table 1: as can be seen, the ILSVRC2012 classes are indeed not fully independent, and certain classes may logically get 'confused' for one another.

4.2 Findings

If for a given image, all models fail to recognize the ground truth class in the top-5, while showing high entropy in their predictions, this may suggest no clear object class is present in the image. Examining $H(\hat{\mathbf{y}})$ for all models, we indeed observe this, and illustrate a particular outlier in Figure 1. Here, due to the cropping procedure, the informative visual content of the image has been removed, and all models were confronted with a white patch.

If for a given image, all models fail to recognize the ground truth class in the top-5, while showing low entropy and consistency in their predictions, this may suggest another object class is more salient than the labeled ground truth. For example, we observe this in Figure 2, for which all models predict the `baseball` class with 1.0 confidence, instead of the (less visually salient) ground truth class `bucket`.

If for a given image class, multiple models consistently have problems recognizing the ground truth class in the top-5, the image class may not visually stand out. Counting top-5 errors per image class, the hardest categories are `velvet` (18/50 images never have a correct top-5 prediction, e.g. see Figure 3), `spotlight` and `letter opener` (in both cases 16/50 images never have a correct top-5 prediction). Indeed, other, more visually consistent and recognizable classes will likely coincide with these classes in images; for example, `velvet` is a material, rather than a truly recognizable object.

If two classes are consistently confused by the models, they may have been synonyms, homonyms, or meronyms. We also performed an analysis of top-1 confusions, counting frequency occurrences of <ground truth, predicted class> pairs over all models. The most common confusions are `laptop` being 'confused' 114 times for `notebook` (synonym), and `maillot, tank suit` being 'confused' 102 times for `maillot` (homonym), the latter confirming an earlier reported 'data bug' in ILSVRC2012 [17]. At the same time, looking at associated prediction confidences, the models do not always consider close semantic concepts to be truly close; e.g., see the strongly varying prediction confidence levels for classes `laptop` and `notebook` in Figures 4 and 5.

[1] https://github.com/calebrob6/imagenet_validation/

(a) Original (b) Cropped

vgg16	vgg19	ResNet50	ResNet101
wallet (0.4160)	doormat (0.3504)	doormat (0.8952)	purse (0.7394)
doormat (0.2878)	purse (0.2684)	pencil box (0.0293)	pencil box (0.0984)
purse (0.1625)	wallet (0.1115)	purse (0.0206)	doormat (0.0975)
pencil box (0.0482)	pencil box (0.0934)	chest (0.0082)	backpack (0.0143)
mailbag (0.0204)	mailbag (0.0402)	mailbag (0.0054)	chest (0.0101)

(c) Predictions

Figure 3: Top-5 classifications for velvet image ILSVRC2012_val_00000433.

(a) Original (b) Cropped

vgg16	vgg19	ResNet50	ResNet101
laptop (0.9592)	laptop (0.9796)	laptop (0.9954)	laptop (0.9984)
notebook (0.0346)	notebook (0.0191)	notebook (0.0042)	notebook (0.0015)
iPod (0.0024)	iPod (0.0004)	space bar (0.0002)	space bar (0.0000)
hand-held computer (0.0011)	desktop computer (0.0002)	computer keyboard (0.0000)	mouse (0.0000)
modem (0.0007)	space bar (0.0001)	mouse (0.0000)	computer keyboard (0.0000)

(c) Predictions

Figure 4: Top-5 classifications for laptop image ILSVRC2012_val_00007373.

(a) Original (b) Cropped

vgg16	vgg19	ResNet50	ResNet101
notebook (0.7222)	notebook (0.7327)	notebook (0.7230)	notebook (0.8161)
laptop (0.1866)	laptop (0.1178)	laptop (0.1689)	laptop (0.1492)
desktop computer (0.0244)	desktop computer (0.0459)	desktop computer (0.0420)	modem (0.0100)
space bar (0.0097)	space bar (0.0243)	space bar (0.0239)	space bar (0.0091)
solar dish (0.0092)	hand-held computer (0.0152)	mouse (0.0059)	desktop computer (0.0041)

(c) Predictions

Figure 5: Top-5 classifications for laptop image ILSVRC2012_val_00002580.

4.3 Good performance vs. visual understanding

Our analysis surfaces various oracle issues, that globally hint at issues with label taxonomies and problems with data encoding and representation. Considering the original setup and context of the ILSVRC2012 data, as an academic benchmark focused on assessing the presence of certain object classes in images, this is not necessarily a problem. As we showed in the previous subsection, many 'mistakes' made by our examined models can be explained by a human and may not be true errors, rather signifying cases in which the oracle may need to be reinterpreted. However, given the interest in deploying well-performing models in real-world scenarios, we want to point out that there still are conceptual discrepancies between very good model performance based on the ILSVRC2012 data, and true visual understanding for safety-critical applications. Models may exist that may yield even better performance than our currently examined models within the ILSVRC2012 context and its representation and evaluation framework, but that may never be acceptable in practical scenarios, e.g. in automated computer vision components for self-driving cars.

ILSVRC2012 is no balanced representation of the real world. Where ImageNet seeks to provide a comprehensive visual ontology, the ILSVRC2012 benchmark made particular benchmark-motivated choices in picking the classes to be recognized. For example, as ILSVRC2012 focused both on general and fine-grained classification, the latter was facilitated with more than 100 out of the 1000 object classes corresponding to sub-species of dogs (e.g. miniature poodle, standard poodle). However, it would be unrealistic to assume that over 10% of our real-world visual observations consider sub-species of dogs.

Image classes in IlSVRC2012 are not independent. However, in the way they are mathematically represented, it is implied they are. With only one ground truth label per image, mathematically, the 'ideal' $\mathbf{y}$ for a given image will be a one-hot encoded vector, with $\mathbf{y}_i = 1.0$ for the i corresponding to the ground truth class, and $\mathbf{y}_i = 0.0$ otherwise. In other words, classes are framed as independent. Thus, mathematically, a miniature poodle would be considered equally far away to a beer bottle as to a standard poodle.

Maximum likelihood criteria will nudge models towards treating the classes as independent. During the training of an ML classification pipeline, the common criterion to optimize for is the likelihood of the ground truth class, which should be maximized. With a single ground-truth label being available per image, the best result in terms of optimization therefore is to have a prediction confidence of 1.0 for a single class (and thus, a probability of 0.0 for other classes), even if multiple classes are present. Thus, while a beach wagon typically contains more than one car wheel, if the first class was the ground truth, optimization is considered to have succeeded better if an ML system classifies beach wagon with 1.0 confidence, thus being 'blind' to the possible presence of car wheels.

Traditional final success assessment ignores prediction confidence. As noticed before, traditional ILSVRC2012 evaluation only cares about the presence of the ground truth class in the top-1 or top-5: whether the predicted probability for a ground truth label is 1.0 or 0.1 does not matter, as long as the class is present. Hence, a

theoretical system that would always output a 0.99 probability of an image containing a `triceratops`, but that would still predict the ground truth class within the top-5 (at a probability < 0.01), would be considered a perfect system according to top-5 accuracy.

The machine may not 'see' what the human annotator saw. Human annotators saw full images at different dimensions, but neural networks need to work with standardized dimensions, leading for an image to be cropped before prediction. As we showed, it is not guaranteed that the object of interest is still visible in the crop.

5 CONCLUSION

In this paper, we have surfaced various oracle issues in ILSVRC2012. That errors and ambiguity exist in ImageNet data has been recognized for long [17, 22, 23], but as an academic benchmark focused on object classification, the presence of the right class in the top-5 has in the past years been seen as sufficient to consider the classification task solved. However, the oracles we currently have at hand, and the representation and evaluation frameworks we currently employ within ML, may be insufficient when considering the deployment of ML components in real-world application scenarios. In safety-critical applications (e.g. self-driving cars), a system with 100% top-1 accuracy on ILSVRC2012 should not exist at all, as it will treat ambiguous recognition problems as unambiguous, and may even have been incentivized to only recognize one class at a time. The computer vision community has only started reacknowledging this very recently (after the reviewing phase of this article), and now is revisiting the need to take possible class hierarchies and the severity of mistakes into account [5].

The higher-level, more conceptual oracle issues we addressed are not trivially identifiable; furthermore, whether they will be harmful in a practical ML application scenario will always require human judgment. Still, with our currently proposed approach, we illustrated how heuristics and comparisons of alternative models can be of assistance. They enable semi-automated analysis approaches, that can cover a wider search range, and that can prioritize potential issues more systematically than a human can achieve alone.

Towards the software engineering community, we want to emphasize that this community has commonly been confronted with situations where human judgment on 'what is correct' is required, yet non-trivial to scale up. In particular, we want to liken the way in which we analyze different model judgments to the philosophy behind differential testing. Taking this comparison further, we foresee interesting future interdisciplinary collaboration opportunities, e.g. towards automated test case generation and prioritization. In this, as we showed, in ML scenarios, the focus of attention may not only be limited to testing and improving software systems, but even the oracles that fundamentally drive them.

REFERENCES

[1] Earl T. Barr, Mark Harman, Phil McMinn, Muzammil Shahbaz, and Shin Yoo. 2015. The oracle problem in software testing: A survey. *IEEE T. Software Engineering* 41, 5 (2015), 507–525.

[2] Denis Baylor, Eric Breck, Heng-Tze Cheng, Noah Fiedel, Chuan Yu Foo, Zakaria Haque, Salem Haykal, Mustafa Ispir, Vihan Jain, Levent Koc, Chiu Yuen Koo, Lukasz Lew, Clemens Mewald, Akshay Naresh Modi, Neoklis Polyzotis, Sukriti Ramesh, Sudip Roy, Steven Euijong Whang, Martin Wicke, Jarek Wilkiewicz, Xin Zhang, and Martin Zinkevich. 2017. TFX: A TensorFlow-Based Production-Scale Machine Learning Platform. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD).*

[3] Kent Beck. 2003. *Test-driven development: by example.* Addison-Wesley Professional.

[4] Houssem Ben Braiek and Foutse Khomh. 2020. On testing machine learning programs. *J. Systems and Software* 164 (2020).

[5] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A. Lord. 2020. Making Better Mistakes: Leveraging Class Hierarchies with Deep Networks. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR).*

[6] François Chollet et al. 2015. Keras. https://keras.io.

[7] Kate Crawford and Trevor Paglen. 2019. Excavating AI: The Politics of Images in Machine Learning Training Sets. https://excavating.ai.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR).*

[9] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR).*

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR).*

[11] Nick Hynes, D. Sculley, and Michael Terry. 2017. The Data Linter: Lightweight, Automated Sanity Checking for ML Data Sets. In *Proceedings of the NIPS Workshop on ML Systems.*

[12] Sanjay Krishnan and Eugene Wu. 2019. AlphaClean: Automatic Generation of Data Cleaning Pipelines. *arXiv preprint arXiv:1904.11827* (2019).

[13] Lei Ma, Fuyuan Zhang, Jiyuan Sun, Minhui Xue, Bo Li, Felix Juefei-Xu, Chao Xie, Li Li, Yang Liu, Jianjun Zhao, and Yadong Wang. 2018. DeepMutation: Mutation testing of deep learning systems. In *Proceedings of the 29th International Symposium on Software Reliability Engineering (ISSRE).*

[14] George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM* 38, 11 (Nov. 1995), 39–41.

[15] Christian Murphy, Gail E. Kaiser, Lifeng Hu, and Leon Wu. 2008. Properties of machine learning applications for use in metamorphic testing. In *Proceedings of the 20th International Conference on Software Engineering & Knowledge Engineering (SEKE).*

[16] Ahn Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR).*

[17] Finn Årup Nielsen. 2018. Linking ImageNet WordNet Synsets with Wikidata. In *Companion Proceedings WWW '18.*

[18] Hung Viet Pham, Thibaud Lutellier, Weizhen Qi, and Lin Tan. 2019. CRADLE: cross-backend validation to detect and localize bugs in deep learning libraries. In *Proceedings of the IEEE/ACM 41st International Conference on Software Engineering (ICSE).*

[19] Caleb Robinson. 2018. How to reproduce ImageNet validation results. https://calebrob.com/ml/imagenet/ilsvrc2012/2018/10/22/imagenet-benchmarking.html

[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Computer Vision* 115, 3 (2015), 211–252.

[21] Sebastian Schelter, Dustin Lange, Philipp Schmidt, Meltem Celikel, and Andreas Grafberger. 2018. Automating large-scale data quality verification. *Proceedings of the VLDB Endowment* 11 (2018). Issue 12.

[22] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR).*

[23] Pierre Stock and Moustapha Cisse. 2018. ConvNets and ImageNet Beyond Accuracy: Understanding Mistakes and Uncovering Biases. In *Proceedings of the 15th European Conference on Computer Vision (ECCV).*

[24] Bob L. Sturm. 2014. The State of the Art Ten Years After a State of the Art: Future Research in Music Information Retrieval. *J. New Music Research* 43, 2 (2014), 147–172. https://doi.org/10.1080/09298215.2014.894533 arXiv:https://doi.org/10.1080/09298215.2014.894533

[25] Xiaoyuan Xie, Joshua Ho, Christian Murphy, Gail Kaiser, Baowen Xu, and Tsong Yueh Chen. 2009. Application of metamorphic testing to supervised classifiers. In *Proceedings of the International Conference on Quality Software (QSIC).* IEEE, 135–144.

[26] Xiaoyuan Xie, Joshua W. K. Ho, Christian Murphy, Gail Kaiser, Baowen Xu, and Tsong Yueh Chen. 2011. Testing and validating machine learning classifiers by metamorphic testing. *J. Systems and Software* 84, 4 (2011), 544–558.

[27] Kaiyu Yang, Klint Qinami, Li Fei-Fei, Jia Deng, and Olga Russakovsky. 2020. Towards Fairer Datasets: Filtering and Balancing the Distribution of the People Subtree in the ImageNet Hierarchy. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*).*

[28] Jie M. Zhang, Mark Harman, Lei Ma, and Yang Liu. 2020. Machine Learning Testing: Survey, Landscapes and Horizons. *IEEE T. Software Engineering* (2020).