Understanding Failures of Deep Networks via Robust Feature Extraction

Sahil Singla*
University of Maryland
ssingla@umd.edu

Besmira Nushi, Shital Shah, Ece Kamar, Eric Horvitz
Microsoft Research
{benushi,shitals,eckamar,horvitz}@microsoft.com

Abstract

Traditional evaluation metrics for learned models that report aggregate scores over a test set are insufficient for surfacing important and informative patterns of failure over features and instances. We introduce and study a method aimed at characterizing and explaining failures by identifying visual attributes whose presence or absence results in poor performance. In distinction to previous work that relies upon crowdsourced labels for visual attributes, we leverage the representation of a separate robust model to extract interpretable features and then harness these features to identify failure modes. We further propose a visualization method aimed at enabling humans to understand the meaning encoded in such features and we test the comprehensibility of the features. An evaluation of the methods on the ImageNet dataset demonstrates that: (i) the proposed workflow is effective for discovering important failure modes, (ii) the visualization techniques help humans to understand the extracted features, and (iii) the extracted insights can assist engineers with error analysis and debugging.

1. Introduction

It is critically important to understand the failure modes of machine learning (ML) systems, especially when they are employed in high-stakes applications. Aggregrate metrics in common use capture summary statistics on failure. While reporting overall performance is important, gaining an understanding of the specifics of failure is a core responsibility in the fielding of ML systems and components. For example, we need to understand situations where a self-driving car will fail to detect a pedestrian even when the system has high overall accuracy. Similarly, it is important to understand for which features misdiagnosis is most probable in chest x-rays even if the model has higher overall accuracy than humans. Such situation-specific insights can guide the iterative process of model development and debugging.

Model performance can be wildly non-uniform for different clusterings of instances and such heterogeneity is not reflected by standard metrics such as AUC or accuracy. For example, it was shown in [19] that a commercial model for emotion detection from facial expressions systematically failed for young children. Buolamwini *et al*. [5] found that gender detection in multiple commercial models had significantly higher error rates for women with darker skin tone. These examples highlight the importance of identifying natural clusters in the data with high failure rates. However, practical problems with these approaches still remain: (a) they require an expensive and time-consuming collection of metadata by humans, and (b) visual attributes that machine learning procedures pay attention to can be very different from the ones humans focus on (see Appendix Section F).

To resolve these issues, we propose to leverage the internal representation of a robust model [25] to generate the metadata. The key property that makes robust representations useful is that the features can be visualized more easily than for a standard model [14, 39]. Our method, named Barlow[1], is inspired from Fault Tree Analysis [23] in safety engineering and uses robust representations as a building block.[2] We demonstrate the results on the ImageNet dataset [12] and find that it reveals two types of failures:

Spurious correlations: A spurious correlation is a feature that is causally unrelated to the desired class but is likely to co-occur with the same class in the training/test data. For example, food is likely to co-occur with plates. However, the absence of the food from a plate image should not result in misclassification (see examples in Figures 1a, 1b and 1e).

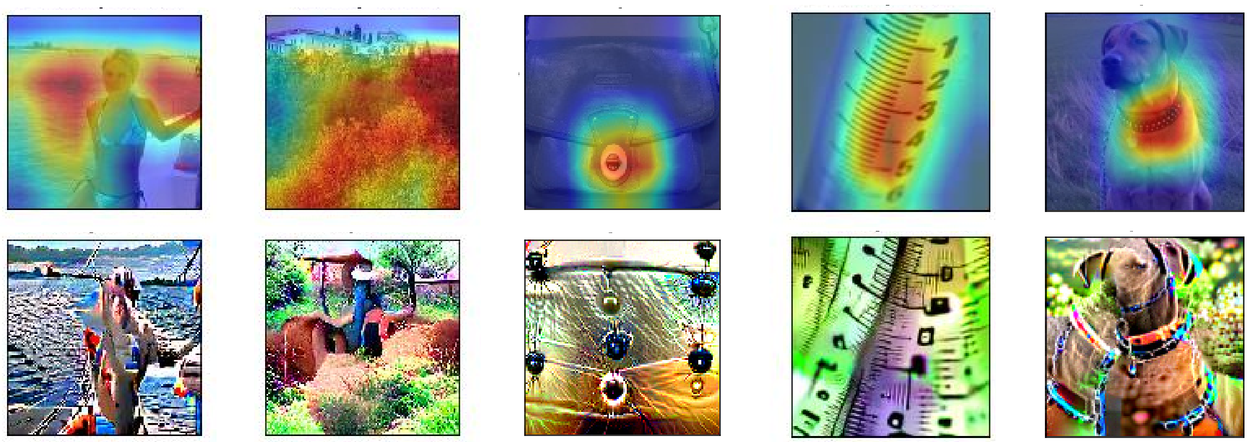
Overemphasized features: An overemphasized feature is a feature that is causally related to the desired class but where the model gives *excessive* importance for classification, disregarding the other relevant features, and is unable to make a correct prediction when that feature is absent from the image. For example, a model may be likely to fail on a purse image if the buckle is absent (see Figures 1c and 1d).

While determining the type of failure (spurious vs. causal but overemphasized feature) and formulating mitigation steps remains a task that depends on human expertise,

*Work carried out during a research internship at Microsoft Research.

[1] In honor of perceptual psychologist, Horace Barlow [3].

[2] Code repository: https://github.com/singlasahil14/barlow



(a) class: maillot
feature: water
error increase: **+9.72%**

(b) class: monastery
feature: greenery
error increase: **+24.84%**

(c) class: purse
feature: buckle
error increase: **+10.94%**

(d) class: syringe
feature: markings
error increase: **+14.99%**

(e) class: rhodesian
feature: dog collar
error increase: **+10.91%**

Figure 1: Failure modes discovered using the proposed methodology for a standard Resnet-50 neural network trained on ImageNet. In the top row, red denotes the region that a specific feature is paying attention to. In the bottom row, we show the image generated by visually amplifying the same feature. We observe that, due to the presence of spurious correlations, the failure rate of the model increases significantly on the relevant class. Additional examples in Appendix Section F.

Barlow assists practitioners in this process by efficiently identifying and providing visualizations of failure modes, showing how input characteristics *correlate* with failures. For example, failures identified in Figure 1 suggest the following interventions: (a) add images of maillot/monastery with diverse backgrounds, Rhodesian Ridgebacks without collar (b) mask overemphasized features (buckle from purse, markings from syringe) in the training set.

In summary, we provide the following contributions:

1. An error analysis framework for discovering critical failure modes for a given model.
2. A feature extraction and visualization method based on robust model representations to enable humans to understand the semantics of a learned feature.
3. A large-scale crowdsourcing study to evaluate the effectiveness of the visualization technique and the interpretability of robust feature representations.
4. A user study with engineers with experience using machine learning for vision tasks to evaluate the effectiveness of the methodology for model debugging.

2. Related work

Feature Visualization, Interpretability, and Robustness: For a trained neural network, feature visualization creates images to either (i) visualize the region that neurons are paying attention to using heatmaps, or (ii) maximize the activation of neurons that we are interested in and visualize the resulting image. Previous works [44, 43, 13, 26, 28, 47, 35, 34, 32, 37, 7, 4] provide evidence that the internal representations of a neural network can capture important semantic concepts. For example, the network dissection method [46, 4, 46] visualizes hidden network units by showing regions that are most activated for that unit and correlates the activation maps with dense human-labeled concepts. In our studies, we observe that activation maps alone are not sufficient for narrowing down the feature concept and that methods for maximizing neuron activation produce noisy visualizations with changes not visible to the human eye (see Appendix Section I). Similar limitations are also discussed by Olah *et al*. [31] in the context of understanding the interpretability of feature visualization methods for standard models. Recent work [39, 2, 25] shows that saliency maps are qualitatively more interpretable for adversarially trained models (when compared to standard training) and align well with human perception. Moreover, Engstrom *et al*. [14] showed that, for robust models (in contrast to standard models), optimizing an image directly to maximize a certain neuron helps with visualizing the respective relevant learned features. We note that disentangled representations [22, 18, 6, 20, 9, 24] based on VAEs [21] can also be used for feature extraction. However, it is difficult to scale these methods to large-scale, rich datasets such as ImageNet.

Failure explanation: In recent years, there has been increasing interest in the understandability of predictions made by machine-learned models [35, 36, 38, 1, 34]. Most of these efforts have focused on *local* explanations, where decisions about single images are inspected [14, 16, 34, 15, 8, 33, 40, 42]. Examining numerous local explanations for debugging can however be time-consuming. Thus, we focus on identifying major failure modes across the entire dataset and presenting them in a useful way to guide model improvement by actions as fixing the data distribution and performing data augmentation. For this, we build upon prior work that discovers generalizable failure modes based on metadata or features [29, 45, 10, 41]. These methods operate typically on tabular data where features are already available [10, 45], on language data with query-definable text operators [41], or on image data where features are collected from intermediate multi-component output and crowdsourcing [29]. In this work, we focus on identifying failure modes for images but with features extracted in an automated manner from learned robust representations.

3. Background and method overview

Let $h : x \rightarrow y$ be a trained neural network that classifies an input image x to one of the classes $y \in Y$. For a cluster of images C in an overall benchmark S (i.e., $C \subset S$), we use the following definitions to quantify failures:

Definition 1 *The error rate of a cluster is the fraction of images in the cluster that are misclassified:*

$$\mathrm{ER}(C) = \frac{\sum_{x \in C} \mathbb{1}_{h(x) \neq y}(x)}{|C|}$$

Definition 2 *The error coverage of a cluster is the fraction of all errors in the benchmark that fall in the cluster:*

$$\mathrm{EC}(C) = \frac{\sum_{x \in C} \mathbb{1}_{h(x) \neq y}(x)}{\sum_{x \in S} \mathbb{1}_{h(x) \neq y}(x)}$$

Definition 3 *The base error rate is the error rate for the whole benchmark, treating it as one cluster:*

$$\mathrm{BER} = \mathrm{ER}(S)$$

We seek to describe failures in a benchmark S with low-dimensional rules using a reduced set of human-interpretable features. For example, for an image recognition model that detects traffic lights, we want to generate failure explanations, such as *"The error rate for detecting traffic lights is 20% higher when an image is captured in rainy weather and low light."*. The failure explanation in this case would be `weather=rainy` $\wedge$ `light_intensity=low`. Such rules slice the data into clusters for which we can report metrics as in Definitions 1 and 2. Ideally, we would like to find clusters that jointly have large error rates and that cover a significant fraction of all errors in the benchmark. These criteria ensure that the explanatory rules will be of sufficient importance and generality. Note that the purpose of these explanations is not to predict failure but rather to provide actionable guidance to engineers via a small set of rules about failures and their indicators. Figure 2 depicts the end-to-end Barlow workflow.

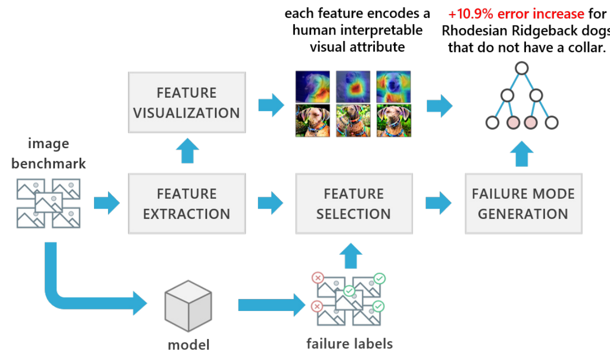


Figure 2: Barlow error analysis workflow. *Feature extraction* and *visualization* are described in Section 4. *Feature selection* and *failure mode generation* in Section 5.

4. Feature extraction and visualization

For each image in the set S, we first construct feature representation F as a vector such that each element of the vector encodes some human-interpretable visual attribute. For this purpose, we use the penultimate layer (i.e., the layer adjacent to the logits layer) of a pretrained robust neural network to extract this feature vector. The robust model is trained via objective (1) for l_2 robustness.

$$\min_{\theta} \ \mathbb{E}_{(x,y)\sim\mathbb{D}} \left[\max_{\|x'-x\|_2 \leq \rho} \ell(h_\theta(x'), y) \right] \qquad (1)$$

Tsipras *et al.* [39] show that for robust models, saliency maps are more interpretable than standard models and align well with human perception. Engstrom *et al.* [14] show that direct maximization of neurons of robust models is sufficient to generate recognizable visual attributes. Such output stands in contrast to the opacity of the visualizations of features generated from standard models for which, even with regularization, visualized features are rarely human interpretable [31] (examples in Appendix Section I). In Sections 8 and 9, we validate the earlier findings about human interpretability via performing studies with human subjects.

Given the findings on the interpretability of robust models, for feature extraction, we use an adversarially trained Resnet-50 model [17] pretrained on ImageNet (using $\rho = 3$). We note that we can perform feature extraction using a robust model even when the model under inspection is not robust. In this case, we can consider the extracted features as attributes of the data and not necessarily as part of the representation employed within the model.

We next describe our methodology for visualizing the neuron in the representation layer of a robust model:
Most activating images: A common approach to visualizing a neuron's sensitivity is to search through the given set of images to find the top-k instances that maximally activate

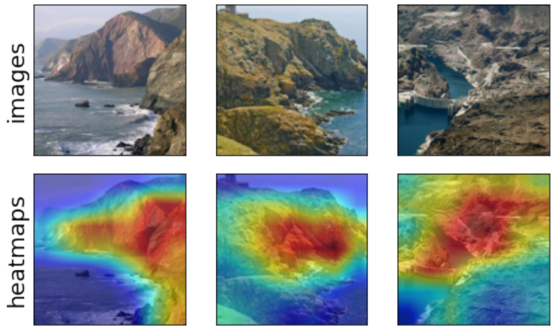


Figure 3: Illustration of heatmaps. From the most activating images (top row) for a neuron, it is not clear if the focus is on sky, water, or ground. Heatmaps (bottom row) resolve the ambiguity. The neuron appears to focus on ground.

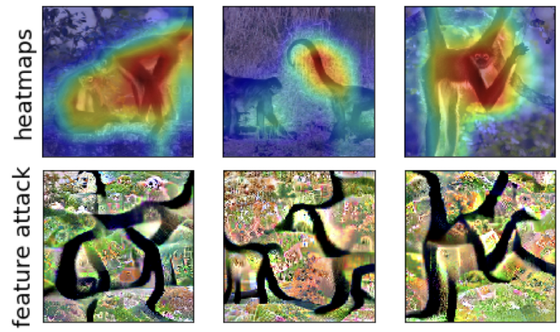


Figure 4: Illustration of feature attack. Heatmaps focus on different attributes in different images: body, tail, and face (left to right). Feature Attack (bottom row) resolves the ambiguity. The neuron appears to focus on the limbs and tail.

the desired neuron. A challenge with this approach is that it does not identify the specific attributes of images that are responsible for the activation [31]. For example, consider the images in the top row of Figure 3 where it is not clear whether the neuron is focused on sky, water or ground.

Heatmaps: To address the aforementioned problem, we propose to use heatmaps based on CAM [47] (second row in Figure 3), as an additional signal for visualization. For a Resnet-50 model, the representation layer is computed via a global average pooling operation to the tensor in the previous layer. We use the index of the desired neuron to retrieve the slice of the tensor from the previous layer at the same index (we refer to this as the *feature map*). Next, we normalize the feature map between 0 and 1 and resize it to match the input size. Details are in Appendix Section B. Figure 3 shows how heatmaps can help resolve the ambiguity.

Feature Attack: In some cases, heatmaps may not be sufficient to resolve ambiguity. For example, in Figure 4, it is not clear from the heatmaps whether the neuron is focused on the body, tail, or face of the monkey. Hence, we propose to maximize the neuron in the representation layer with respect to the original image (based on Engstrom *et al*. [14]). We observe that the limbs and tail (on top of green background) are amplified and this resolves the ambiguity. We call the resulting visualization, a *feature attack*. More details are provided in Appendix Section C.

We use the top-6 most activating images, corresponding heatmaps, and feature attack images to generate visualizations of the desired feature (Examples in Appendix Section G). We showcase the importance of these visualizations for interpretability via a study detailed in Section 8.

5. Failure mode generation

We start with a set of features extracted from images using a robust model. Then, we explore the failure modes of a neural network (not necessarily the robust model) identified via consideration of the ground-truth labels. Our goal is to identify clusters with high error rates. In pursuit of failure analyses that can be understood with ease, we generate decision trees trained to predict failures (see Figure 2), determined by checking whether the model prediction is equal to the image label.

We note that there are challenges with relying on decision trees for explanations. Large numbers of features can make finding a good split difficult due to the curse of dimensionality. From a usability perspective, failures for different classes can vary greatly and practitioners may be interested in understanding them separately [29]. Moreover, psychological studies provide evidence that people can hold in memory and understand only a limited number of chunks of information at the same time [27, 30, 11]. These challenges motivate designs for Barlow and other human-in-the-loop methodologies that avoid complex representations, such as large trees. Thus, we take the following steps:

Class-based failure explanations: We focus analyses and results on two types of class-based groupings: (i) *prediction groupings* and (ii) *label groupings*. For example, *prediction groupings* for the class "goldfish", contain all images for which the model under inspection predicts "goldfish". A failure analysis for this group enables us to understand false positives. *Label groupings* for the class "goldfish", contain all images for which the ground truth in ImageNet is "goldfish" regardless of the model's prediction. A failure analysis for this class provides insights about false negatives.

Feature selection: To reduce dimensionality, we compute the mutual information between each feature and failure labels in the group, and select the top-20 features, sorted by mutual information values. Mutual information estimates the information gained via the feature on the failure label variable in addition to its prior (i.e., the base error rate). Using mutual information for feature selection aligns with our goal of building decision trees with the features, as the same

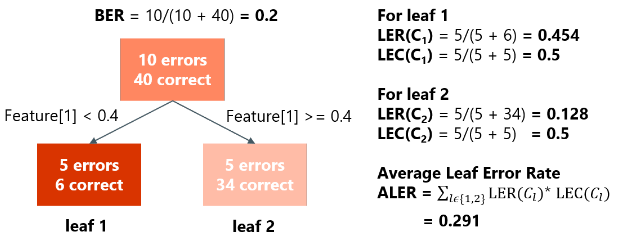


Figure 5: Illustration of ALER metric. For both leaf nodes, precision and recall are 0 since $\mathrm{ER} < 0.5$ however leaf 1 is more important for failure mode discovery.

metric is used to estimate the value of splits in the tree.
Truncation and rule generation: All trees are truncated to a low depth (≤ 3). We choose to explain failure modes by using paths in the tree as summary rules that have an error coverage higher than a given threshold τ and an error rate of at least $\mathrm{BER} + \delta$. Both δ and τ are adjustable. This procedure is described in Appendix Algorithm 1.

5.1. Automatic evaluation of decision tree

Traditional metrics for evaluating the quality of a decision tree for a classification problem, such as accuracy, precision, and recall, are insufficient when the model is used for description and explanation. Ideally we would want to consider failure modes that include all possible failures in a benchmark. This goal is challenging because of (i) incompleteness in the feature set, (ii) difficulties in finding failure modes that generalize well for many examples at the same time, and, most importantly, (iii) certain failures may happen very rarely in the benchmark. Therefore, we focus on the explanatory properties of failure analyses.

For a given leaf node l, we can use definitions 1 and 2 to define its *leaf error rate* $\mathrm{ER}(C_l)$, and *leaf error coverage* $\mathrm{EC}(C_l)$. C_l denotes the cluster of data that falls into leaf l. For a decision tree T, we then compute the following metric as the *average leaf error rate* (ALER).

Definition 4 ALER *of a tree T is the average error rate across all leaves weighed by the respective error coverage.*

$$\mathrm{ALER}(T) = \sum_{l \in \, leaves(T)} \mathrm{ER}(C_l) \times \mathrm{EC}(C_l)$$

Per the definition of the leaf error coverage, we have $\sum_{l \in \, leaves(T)} \mathrm{ER}(C_l) = 1$ because all failure instances will be covered by exactly one leaf.

As an example, consider the simple 1-depth decision tree given in Figure 5. Since the leaf error rate is less than 0.5 for both leaf nodes, tree precision and tree recall are both zero. However, since the error rate of leaf 1 is significantly higher than the base error rate ($0.454 >> 0.2$), leaf 1 is important as a failure mode description (and more predictive than leaf 2), which is exactly the notion we seek to capture for describing high concentrations of error.

The key property that makes ALER useful is that if it is equal to some quantity q, then there is at least one leaf in the decision tree with leaf error rate greater than q:

Proposition 1 *For a tree T with $\mathrm{ALER}(T) = q$, there exists at least one leaf l, with leaf error rate $\mathrm{ER}(C_l) \geq q$.*

The proposition follows from the fact that all weights (i.e., leaf error coverage) are less than or equal to 1. For the decision tree in Figure 5, $\mathrm{BER} = 0.291$, which is greater than the base error rate 0.2 by a margin of 0.091. This signals the presence of a leaf with error rate of at least 0.291 which we know is leaf 1, $\mathrm{ER}(C_1) = 0.454$.

Since the root node of the tree already comes with a prior on the error rate (BER), we are interested in how much more value the discovered failure modes in the tree add when compared to the root. Thus, for the automated evaluation we use $\mathrm{ALER} - \mathrm{BER}$ to measure the increase in error and discrepancy that the tree explains.

This metric also suggests that leaves with higher value of $\mathrm{ER}(C_l) \times \mathrm{EC}(C_l)$ contribute more to ALER and are thus more important for explaining failures. This leads to the following metric for ranking nodes for failure explanation:

Definition 5 *The Importance Value i.e $\mathrm{IV}(C_l)$ of a leaf node l in a tree T is defined as:*

$$\mathrm{IV}(C_l) = \mathrm{ER}(C_l) \times \mathrm{EC}(C_l)$$

6. Failure modes discovered by Barlow

We now describe some failure modes discovered by Barlow when analyzing a standard Resnet-50 model, using a robust Resnet-50 model for feature extraction (both trained on ImageNet). We use the ImageNet training set (instead of the validation set) for failure analysis due to the larger number of image instances per class (1300 vs. 50 in the validation set). Note that, from a methodology perspective, practitioners may apply the Barlow workflow to any benchmark with ample data and failures, but the failure modes that they discover may vary depending on the nature of the benchmark. For ease of exposition, all decision trees have depth one. We selected the leaf node with highest $\mathrm{IV}(C_l)$ and visualized the feature responsible for the split. More examples of failure modes are in Appendix Section F.1 (using a standard model) and Section F.2 (using a robust model).
Label grouping: In Figure 6, the top row shows the most highly activating images for a feature identified important for failure. The heatmap and feature attack provide strong evidence that the feature is paying attention to the buckle on purses. The bottom row shows randomly selected failures in this leaf node. We do not observe a buckle in any of these images even though they have ground-truth labels

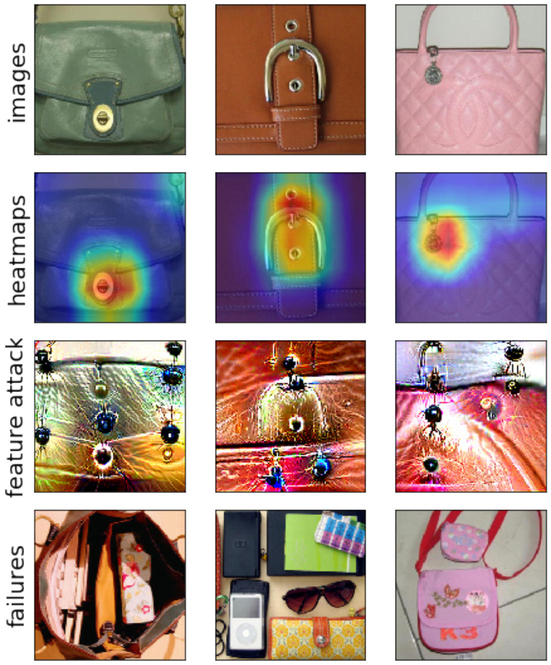


Figure 6: For images with **label purse**, BER = 0.3085. When feature[1456] $<$ 0.3641, error rate increases to 0.4179 (**+10.94%**) and a fraction of 0.6409 of all failures are in this node. ALER = 0.3433.

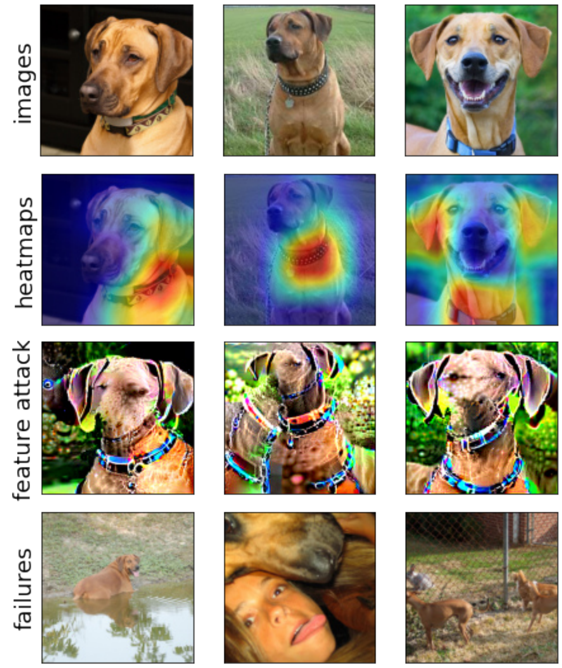
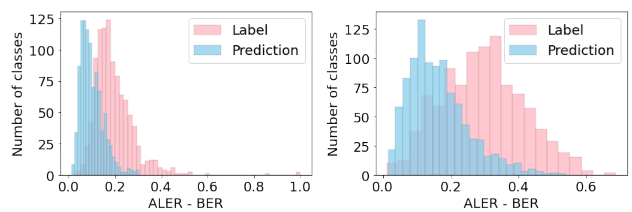


Figure 7: For images with **prediction Rhodesian Ridgeback**, BER = 0.2093. When feature[1634] $<$ 1.3779, error rate increases to 0.3184 (**+10.91%**) and a fraction of 0.4561 of all failures are in this node. ALER = 0.2337.



(a) Standard model

(b) Robust model

Figure 8: Comparison between grouping strategies using a decision tree of depth 3.

of purse indicating the feature (while correlated, not causal) is important for making correct predictions. In other words, it appears that, whenever the purse image does not include a buckle, it is more difficult for the model to predict it correctly as a purse (error increases by 10.94%) and the prediction is more likely to be a false negative.

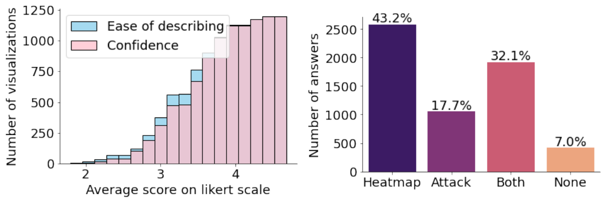
Prediction grouping: Figure 7 displays an example where the heatmap and feature attack provide strong evidence that the feature indicates a dog collar. In the bottom row (i.e randomly selected failure images), we do not observe a dog collar in any of these images even though the model predicts a Rhodesian Ridgeback for all of them. This indicates that the feature (correlated, not causal) is important for making correct predictions. In other words, whenever the model predicts Rhodesian Ridgeback, and the image does not contain a dog collar, the prediction is more likely to be a false positive (error increases by 10.91%).

7. Experiments with automated evaluation

We now report on a study of factors that influence the effectiveness of error analysis: decision tree depth, robustness of model, and grouping strategy. We train decision trees with depths of 1 and 3 for each model and grouping strategy. For evaluating a decision tree, we use the metric ALER $-$ BER as defined in Section 5.1. We also select the leaf with highest importance value $\mathrm{IV}(C_l)$ for each decision tree and evaluate whether the cluster of data in this leaf satisfies the two conditions: $\mathrm{ER}(C_l) > \mathrm{BER} + \delta$ and $\mathrm{EC}(C_l) > \tau$, with $\delta = 0.1$ and $\tau = 0.2$. In Appendix Table 1, we report for each model, grouping strategy, and tree depth the fraction of such *valid leaves* across all 1000 classes that satisfy these conditions.

In summary, we make the following observations:

- Grouping by ground-truth labels results in better decision trees (by ALER $-$ BER score) as compared to prediction grouping for both standard and robust models (Figure 8), and also for decision trees with different depths.
- Failure explanation for a robust model results in significantly better score compared to the standard model for



(a) Questions 3 and 4

(b) Question 5

Figure 9: Cumulative distribution of answers to Questions 3, 4, 5 in the crowd study.

both grouping strategies and depths of decision tree (See Appendix Figures 16, 19).

- Increasing depth significantly improves the score for both models as shown in Appendix Figures 17 and 20.
- For the standard model, on all trees (except prediction grouping and depth=1 where it is 0.211) the fraction of classes with at least one valid leaf is at least 0.596. For the robust model, on all trees the fraction of classes with at least one valid leaf is at least 0.787.

8. Crowd study on feature interpretability

To understand the effectiveness of feature visualizations, we conducted a crowdsourcing study using Amazon Mechanical Turk. For both grouping strategies (prediction and label), we selected the top, middle, and bottom 20 classes (total 60 x 2) based on the error rate of the robust Resnet-50 model and selected the top 10 features with the highest mutual information on failure, resulting in 1200 visualizations and 5971 human answers. In total, we had 312 unique workers, each completing an average of 19.14 tasks. All visualizations were evaluated by five workers, except 29 for which we only acquired four assessments. Workers were paid \$0.5 per hit, with an average salary of \$12 per hour. Anonymized data from this study is available in the associated code repository.

For each visualization, workers were shown three sections: A (most activating images), B (heatmaps), and C (feature attack), and asked to answer the set of questions in Figure 65 in the Appendix Section G. The questions were designed with two goals: (1) to collect human-generated feature descriptions, and (2) to evaluate the ease with which workers can understand and describe the features.

Figure 9a shows the cumulative distribution of answers to Questions 3 (Ease of understanding) and 4 (Confidence). We observe that workers are able to describe visualizations with confidence and ease (average likert scale > 3 for both) for a large number of features (> 800). Figure 9b shows workers' preferences of heatmaps versus feature attacks (Question 5). In response to a question about the most useful views, 43.18% answers (out of 5971) report Section B (heatmap) as most useful, 17.67% report Section C (feature attack), and 32.14% report both Section B and C. Only 7% of the answers reported None. These results provide evidence that heatmaps are most valuable in contributing to the understanding of the meaning of features, and that feature attack visualizations are also valuable individually and together with heatmaps (examples in Sections G.3 and G.4). Importantly, only 0.92% of all 1200 features received None as the majority vote from all 5 workers. This provides further evidence that the methodology is effective in explaining a large number of features. Note that since crowd workers do not routinely visualize and debug models using such visualizations, our results are likely to be a lower bound on the true effectiveness of the visualizations for engineers.

For further details, Appendix Section G shows an analysis on the agreement scores between textual descriptions of different workers as well as concrete examples that workers found easy or difficult to describe.

9. Study with machine learning practitioners

To evaluate the usefulness of Barlow for error analysis and debugging, we conducted user studies with 14 ML practitioners at Microsoft. Participants were recruited via four mailing lists on the topics of "Machine Learning" and "Computer Vision". Participation requirements included previous experience in applying machine learning on vision tasks. A summary of roles and years of experience is shown in Table 8 in the Appendix Section H.

Study Protocol: Each study lasted one hour and started with a description of the Barlow workflow and terminology as shown in Figure 2. We asked participants to imagine facing a situation as follows:

"*We will conduct error analysis for a classification model (robust Resnet-50) trained on ImageNet. We ask you to imagine that you are part of a team that will deploy this in production and wants to understand where the model is incorrect and identify action items upon failures.*"

All participants inspected two groupings. The first grouping was randomly assigned from a set of five preselected groupings where the most important features for failure explanation were rated as "easy to describe" by crowd workers to facilitate onboarding (exact assignment in Table 9, Appendix Section H). The second grouping was selected randomly. In each case, participants were presented with a decision tree (of depth 2) describing the failure modes. For each node, they could see the error rate and coverage, the instances in the node, and the visualization of the feature responsible for the split. To collect feedback on their experience, we asked the following questions:

Q1: *Can you describe the cluster of images defined by this feature?* All participants were able to describe the features of at least one of the two groupings they inspected. 10 out of 14 participants were able to describe features of both group-

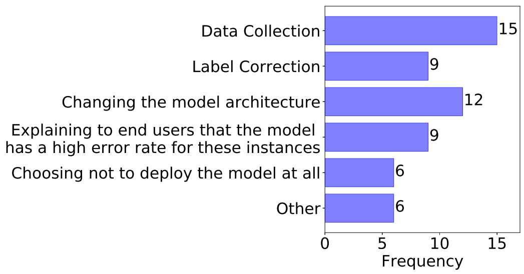


Figure 10: Answers of practitioners on identifying action items for mitigating failures discovered by Barlow.

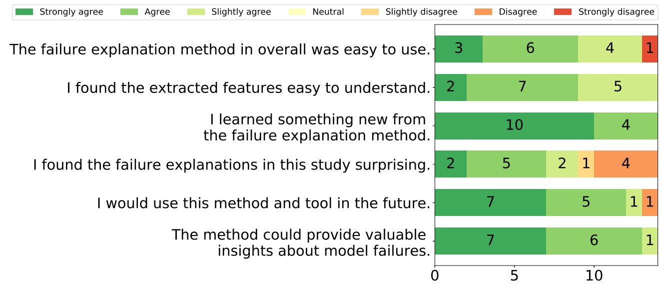


Figure 11: Agreement scores of participants on general evaluation of Barlow.

ings. The four participants who could not describe the features of one grouping faced one or both of the following challenges: the feature represented two or more concepts (e.g., sky and wires), or participants had a difficult time understanding the sensitivity of the feature's visual appearance to its activation value. In such cases, they preferred to see more than five examples for refining their hypothesis.

Q2: *Is the feature necessary for the task or do you think it is a spurious correlation?* Participants were able to identify several surprising spurious correlations (e.g., "*This feature looks like a hair detector, should not be used for Seat Belts.*" - P12, "*Seems like the model can do well only when the photos of Water Jugs are taken either professionally or on a clean background.*" - P17). Similarly, they could identify when the failure modes were related to necessary but not sufficient features (e.g., "*The model is focusing on the face of the dog. We might also need to make the model look at the legs, hair length, hair color, etc.*" - P7).

Q3: *What action items would you take for mitigating the errors for this class?* Overall, participants reported that the identified failure explanations gave them ideas about how to continue with mitigating the errors, given sufficient time and resources. Data collection was the most popular action item (Figure 10) either for mitigating lack of data diversity or addressing statistical biases (e.g. "*I would check if most Tiger Cats in the training data have green background and if so, add more diverse photos.*" - P5). Changes in model architecture were most often related to increasing the capacity of the model (e.g., "*The model is using the same features for detecting zebra patterns and spatulas, perhaps it's best to use more than one feature for this.*" - P5).

User satisfaction scores: The study ended with a survey on whether participants agreed with six statements, completed in private (Figure 11). We observe overall enthusiasm about the usefulness of the workflow and its ability to surface important failures (e.g. "*The approach can help make DNNs more explainable, making such analysis mandatory before deploying a solution. A step further could be to take a similar test with humans to ensure the model has learned explainable features, before we ship the model.*" - P7).

10. Conclusion and future work

We described proposed methods and a workflow aimed at providing insights about critical failures of machine learning models by using features extracted from robust representations. Beyond developing and refining the approach, we performed two sets of studies with human subjects, focusing on the interpretability of features and on the usefulness of the error analysis in realistic settings. The studies provide evidence that practitioners can effectively leverage the methods to interpret the features and to identify most significant clusters in data with errors due to issues, such as systematic spurious correlations. We believe that the error analyses and visualizations embodied in Barlow show promise for supporting cycles of iterative improvement that can identify and mitigate failures without requiring expensive manual metadata collection.

Directions for refining Barlow include extending the range of actions suggested automatically to practitioners and providing richer interactive capabilities. Interactive explorations can include hypothesis testing via the consideration of counterfactuals that make specific, understandable changes to the data and models. Another enabling direction centers on extending the current correlational analyses to causal studies. Extensions that establish causal influences on failure modes will be valuable for debugging non-robust models with robust features. To fairly evaluate counterfactual analyses, it is necessary to establish notions of causal accuracy that reward a model only when it is accurate for the right reasons, as reducing model access to features that are highly predictive but spurious (e.g. food for detecting plates) often will decrease standard measures of accuracy.

Acknowledgments: We thank Parham Mohadjer for helping with our studies with human subjects, Nathan Cross for helpful discussions on extending Barlow to radiology images, and Soheil Feizi for providing useful feedback on the paper and throughout the project. We thank all participants in our studies; the feedback received helped us to refine the methods. Finally, we acknowledge the support provided by Arda Deniz Eden, our newest team member.

References

[1] Julius Adebayo, J. Gilmer, M. Muelly, Ian J. Goodfellow, M. Hardt, and Been Kim. Sanity checks for saliency maps. In *NeurIPS*, 2018. 2

[2] Zeyuan Allen-Zhu and Yuanzhi Li. Feature purification: How adversarial training performs robust deep learning. *arXiv preprint arXiv:2005.10190*, 2020. 2

[3] Horace Barlow. Single units and sensation: A neuron doctrine for perceptual psychology. *Perception*, 1972. 1

[4] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, 2020. 2

[5] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *FAT*, 2018. 1

[6] Christopher P. Burges, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. In *NeurIPS*, 2017. 2

[7] Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. Activation atlas. *Distill*, 2019. https://distill.pub/2019/activation-atlas. 2

[8] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. In *International Conference on Learning Representations*, 2019. 2

[9] Ricky T. Q. Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *NeurIPS*, 2018. 2

[10] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. Slice finder: Automated data slicing for model validation. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1550–1553. IEEE, 2019. 3

[11] Nelson Cowan. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and brain sciences*, 24(1):87–114, 2001. 4

[12] J. Deng, Wei Dong, R. Socher, L. Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 1

[13] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 2

[14] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations, 2019. 1, 2, 3, 4

[15] Yash Goyal, Amir Feder, U. Shalit, and Been Kim. Explaining classifiers with causal concept effect (cace). *ArXiv*, abs/1907.07165, 2019. 2

[16] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *ICML*, 2019. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015. 3

[18] I. Higgins, Loïc Matthey, A. Pal, C. Burges, Xavier Glorot, M. Botvinick, S. Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017. 2

[19] Ayanna Howard, Cha Zhang, and Eric Horvitz. Addressing bias in machine learning algorithms: A pilot study on emotion recognition for intelligent systems. In *IEEE Workshop on Advanced Robotics and its Social Impacts*, 2017. 1

[20] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. ICML, 2018. 2

[21] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 2

[22] Tejas D. Kulkarni, Will Whitney, Pushmeet Kohli, and Joshua B. Tenenbaum. Deep convolutional inverse graphics network. In *NIPS*, 2015. 2

[23] W. S. Lee, D. L. Grosh, F. A. Tillman, and C. H. Lie. Fault tree analysis, methods, and applications a review. *IEEE Transactions on Reliability*, R-34(3):194–203, Aug 1985. 1

[24] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *ICML*, 2019. 2

[25] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 1, 2

[26] Aravindh Mahendran and A. Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120:233–255, 2016. 2

[27] George A Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956. 4

[28] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. In *ICML Workshop on Visualization for Deep Learning*, 2016. 2

[29] Besmira Nushi, Ece Kamar, and Eric Horvitz. Towards accountable AI: hybrid human-machine analyses for characterizing system failure. In Yiling Chen and Gabriella Kazai, editors, *Proceedings of the Sixth AAAI Conference on Human Computation and Crowdsourcing, HCOMP*, pages 126–135. AAAI Press, 2018. 3, 4

[30] Klaus Oberauer and Laura Hein. Attention to information in working memory. *Current Directions in Psychological Science*, 21(3):164–169, 2012. 4

[31] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization. 2, 3, 4

[32] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 2018. https://distill.pub/2018/building-blocks. 2

[33] Matthew O'Shaughnessy, Gregory Canal, Marissa Connor, Mark Davenport, and Christopher Rozell. Generative causal explanations of black-box classifiers. In *NeurIPS*, 2019. 2

[34] R. R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, D. Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128:336–359, 2019. 2

[35] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*, 2014. 2

[36] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. In *ICML Workshop on Visualization for Deep Learning*, 2017. 2

[37] Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Visualizing the impact of feature attribution baselines. *Distill*, 2020. https://distill.pub/2020/attribution-baselines. 2

[38] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, 2017. 2

[39] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *ICLR*, 2018. 1, 2, 3

[40] Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review, 2020. 2

[41] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. Errudite: Scalable, reproducible, and testable error analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 747–763, 2019. 3

[42] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David I. Inouye, and Pradeep D. Ravikumar. On the (in)fidelity and sensitivity of explanations. In *NeurIPS*, 2019. 2

[43] Jason Yosinski, Jeff Clune, Anh Mai Nguyen, Thomas J. Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. In *ICML Deep Learning Workshop*, 2016. 2

[44] Matthew D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 2

[45] Jiawei Zhang, Yang Wang, Piero Molino, Lezhi Li, and David S Ebert. Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. *IEEE transactions on visualization and computer graphics*, 25(1):364–373, 2018. 3

[46] Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. Interpreting deep visual representations via network dissection. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2131–2145, 2018. 2

[47] B. Zhou, A. Khosla, Lapedriza. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4