



Understanding imbalanced data: XAI & interpretable ML framework

Damien Dablain¹ · Colin Bellinger² · Bartosz Krawczyk³ · David W. Aha⁴ · Nitesh Chawla¹

Received: 26 May 2023 / Revised: 15 August 2023 / Accepted: 3 October 2023 /
Published online: 16 January 2024
© The Author(s) 2024

Abstract

There is a gap between current methods that explain deep learning models that work on imbalanced image data and the needs of the imbalanced learning community. Existing methods that explain imbalanced data are geared toward binary classification, single layer machine learning models and low dimensional data. Current eXplainable Artificial Intelligence (XAI) techniques for vision data mainly focus on mapping predictions of specific *instances* to inputs, instead of examining *global* data properties and complexities of entire classes. Therefore, there is a need for a framework that is tailored to modern deep networks, that incorporates large, high dimensional, multi-class datasets, and uncovers data complexities commonly found in imbalanced data. We propose a set of techniques that can be used by both deep learning model users to identify, visualize and understand class prototypes, sub-concepts and outlier instances; and by imbalanced learning algorithm developers to detect features and class exemplars that are key to model performance. The components of our framework can be applied sequentially in their entirety or individually, making it fully flexible to the user’s specific needs (https://github.com/dd1github/XAI_for_Imbalanced_Learning).

Keywords Explainable AI · Interpretable ML · Imbalanced learning · Deep learning

1 Introduction

Convolutional neural networks (CNNs) are increasingly being used in high-stakes fields such as medical diagnosis (Tjoa & Guan, 2020) and autonomous driving (Levinson et al., 2011). Yet, their decisions can be opaque, which makes it challenging for machine learning (ML) algorithm developers to diagnose and improve model performance. The black-box nature of neural networks has spawned the field of explainable Artificial Intelligence (XAI), which seeks to develop techniques to interpret and explain models to increase trust, verifiability, and accountability (Gunning & Aha, 2019). The term Interpretable Machine Learning (IML) is sometimes used to distinguish it from methods that offer an “explanation”, which has a rich

Editor: Vu Nguyen, Dani Yogatama.

Extended author information available on the last page of the article

history in the social sciences as involving human interaction and human subject studies to evaluate the quality of an explanation (Miller, 2019; Hoffman et al., 2018). We use both terms (XAI and IML) in this paper, since no clear, commonly agreed upon definition of *explanation* and *interpretation* exists (Linardatos et al., 2020).

The perceived need to enhance the interpretability of deep learning models has resulted in a number of techniques that are specifically targeted to fields that use ML, such as medicine (Bruckert et al., 2020), air traffic control (Xie et al., 2021), finance (Chen et al., 2018), and autonomous driving (Levinson et al., 2011). However, there is a paucity of IML techniques that have been explicitly adapted for imbalanced data.

Interpretation is critical to both imbalanced learning and IML; although both fields have approached it from different perspectives. IML has generally focused on *model* interpretability; whereas imbalanced learning has sought to better understand *data* complexity. In contrast, imbalanced learning has typically sought to understand the interplay of class imbalance with overlap, sub-concepts and data outliers because imbalanced data can exacerbate model latent feature entanglement, class overlap, and the impact of noisy instances on classifiers (Denil & Trappenberg, 2010; Prati et al., 2004; Jo & Japkowicz, 2004). In addition, many IML techniques usually seek to explain model decisions with respect to specific *instances*; whereas imbalanced learning is generally concerned with the global properties of *entire classes*.

In this work, we combine facets of both fields into a single framework to better understand a CNN's predictions with respect to imbalanced data. We do not develop a single method to improve the interpretability of complex, imbalanced datasets. Rather, we propose a framework and suite of tools that can be used by both model developers and users to better understand imbalanced data and how a deep network acts on it.

In this paper, we make the following research contributions to the field of imbalanced learning:

- *Framework for understanding the high-dimensional imbalanced data.* Many existing imbalanced learning techniques that assess data complexity are designed for binary classification on low-dimensional data and shallow ML models. Because we use the low-dimensional latent representations (Sect. 3.1) learned by a CNN, we are able to provide a suite of tools (Sect. 3) that efficiently visualize specific concepts that are central to imbalanced learning: class prototypes and sub-concepts (Sect. 3.2) and class overlap (Sect. 3.4).
- *Predict relative false positives by class during inference with training data.* We show that the likely classes that will produce the most false positives during inference for a given reference class can be predicted from *training* data (Sect. 3.3).
- *Class saliency color visualizations.* Existing IML methods display black and white heatmaps of pixel saliency for single dataset instances. We, instead, visualize the most salient *colors* used by CNN models to identify *entire classes*. Similar to IML saliency methods, we use the gradient of individual instances to map decisions to input pixels; however, we aggregate this information efficiently across all instances in large datasets by using color prototypes and latent feature embeddings (Sect. 3.5).

2 Background and related work

In this section, we introduce the guiding principles in IML and imbalanced learning that animate our framework:

- *Data is an important element of model understanding.* Advances in deep learning have been built, in part, on access to large amounts of data. Therefore, it is critical to understand how the model organizes data into low dimensional representations used for classification.
- *Need for global data complexity insights to explain deep networks.* Many current IML methods are instance-specific; whereas imbalanced learning explanation requires intuition about global (class) characteristics.
- *CNN texture bias as interpretation.* The perceived texture bias of CNNs can be used to extract informative global, class-wise insights.

We also discuss the prior work that inspires our research and how our approach differs from previous methods.

2.1 Centrality of data to deep learning and class imbalance understanding

Deep learning has shown significant progress in the past decade due, in part, to the ubiquity of low cost and freely available data (Marcus, 2018). Deep networks are typically trained on thousands and even millions of examples to minimize the average error on training data (empirical risk minimization) (Zhang et al., 2018). As the size and complexity of modern datasets grow, it is increasingly important to provide model users and developers vital information and visualizations of representative examples that carry interpretative value (Bien & Tibshirani, 2011). In addition, when deep networks fail on imbalanced data, it is not always intuitive to diagnose the role of data complexity on classifier performance (Kabra et al., 2015).

In imbalanced learning, several studies have assessed the complexity of the data used to train machine learning models; however, many of these studies were developed for small scale datasets used in single layer models. Barella et al. (2021) provide measures to assess the complexity of imbalanced data. Their package is written for binary classification and is based on datasets with 3000 or fewer instances and less than 100 features. Batista et al. (2004) determined that complexity factors such as class overlap are compounded by data imbalance. Their study was performed with respect to binary classification on datasets with 20,000 or fewer examples and 60 or fewer features. Their conclusion that class overlap is a central problem when studying class imbalance was confirmed by Denil and Trappenberg (2010), Prati et al. (2004) and García et al. (2007). Rare instances, class sub-concepts and small disjuncts can also exacerbate data imbalance, add to data complexity and contribute to classifier inaccuracy (Jo & Japkowicz, 2004; Weiss, 2004; Aha, 1992). Ghosh et al. (2022) explore the use of geodesic and prototype-based ensemble to preserve interpretability on a synthetic dataset, a non-public dataset with 496 features and a public dataset with 13 features and less than 1000 instances, although their visualizations focus solely on decision boundaries.

Therefore, understanding data complexity, including class overlap, rare, border and outlier instances, is critical to improving imbalanced learning classifiers. This is especially important in deep learning, where opaque models trained with batch processing may obscure underlying data complexity (Ras et al., 2022; Burkart & Huber, 2021). Unlike prior work, which explained data complexity by examining model inputs, we explain data complexity via the latent features learned by a model. These low-dimensional representations are the raw material used by the final classification layer of CNNs to make their predictions.

2.2 Global (class) vs. instance level interpretation

Several studies have shown that interpretation is critical to machine learning model user satisfaction and acceptance (Teach & Shortliffe, 1981; Ye & Johnson, 1995). It is also important for model developers for diagnostic and algorithm improvement purposes. Explanation is central to both IML and imbalanced learning; however, these fields approach it in different ways.

In IML, great strides have been made to increase model interpretability by describing the inner workings of models and justifying how or why a model developed its prediction (post-hoc explanation) (Kenny et al., 2021). In general, IML techniques can roughly be divided into four groups.

First, there are methods that explain a model's predictions by attributing decisions to inputs, including pixel attribution through back-propagation (Simonyan et al., 2013; Selvaraju et al., 2017; Sundararajan et al., 2017; Zhou et al., 2016). These methods generally work on *single* data instances and do not provide an overall view of class homogeneity, sub-concepts, or outliers (Huber et al., 2021). For example, CAM (Zhou et al., 2016), GRAD-CAM (Selvaraju et al., 2017) and pixel propagation (Simonyan et al., 2013) all highlight the most important pixels that a model uses to predict a single instance of a class. In contrast, our methods show the most relevant feature embeddings and colors for entire classes (i.e., all instances in a class).

Second, explanations by example provide evidence of the model's prediction by citing or displaying similarly situated instances that produce a similar result or through counter-factuals—instances that are similar, yet produce an opposite or adversary result (Lipton, 2018; Keane & Kenny, 2019; Artelt & Hammer, 2019, 2020; Mothilal et al., 2020). Like pixel attribution methods, this approach only provides explanations for single instances or predictions.

Third, there are methods that explain a complex neural network by replacing, or modifying, it with a simpler model. These approaches include local interpretable model explanations (LIME) (Ribeiro et al., 2016), Shapley values (occlusion-based attribution) (Shapley, 1953), the incorporation of the K-nearest neighbor (KNN) (Fix & Hodges, 1989; Cover & Hart, 1967) algorithm into deep network layers (Papernot & McDaniel, 2018), and decision boundary visualizations. Both LIME and Shapley values can be computationally expensive because they involve repeated forward passes through a model (Achtibat et al., 2022). Methods that visualize decision boundaries, such as DeepView (Schulz et al., 2019), often rely on another model [e.g., UMAP (McInnes et al., 2018)] for dimensionality reduction and select a subset of a dataset to produce scatter plots. In contrast, our methods globally utilize a CNN's internal representations for all instances in a training or test set to visualize classes that overlap (including the percentage of overlap), display class sub-concepts, and the most relevant colors that a CNN uses to distinguish an entire class.

Finally, there are IML methods that extract rules learned by a model (Zilke et al., 2016) and the features or concepts represented by individual filters or neurons (Gilpin et al., 2018).

In summary, many existing IML methods offer interpretations for single instances and do not describe the broad class characteristics learned, or used, by a neural network to arrive at its decision. By contrast, in imbalanced learning, the focus of most explanatory methods has been on the global properties of data and classes within a dataset, including the inter-play of class imbalance and data complexity factors, such as class overlap, sub-concepts and noisy examples.

In imbalanced learning, Napierala and Stefanowski partitioned *minority* classes into instances that were homogeneous (safe), residing on the decision boundary (border), rare, and outliers (Napierala & Stefanowski, 2016). We extend their method to *both* majority and minority classes and use a model's *latent* representations to identify instance similarity based on the local neighborhood, instead of using the input space.

2.3 CNN texture bias as explanation

Recent work has demonstrated that CNNs emphasize texture over shape for object recognition tasks (Geirhos et al., 2018; Baker et al., 2018; Hermann et al., 2020). A precise definition of texture remains elusive (Haralick, 1979). Due to the difficulty of precisely defining texture, we focus on one of its properties—color or chromaticity of a region. We use a CNN’s color bias as *explanation*. As discussed in more detail in Sect. 3.5, we combine both saliency maps and pixel aggregation to reveal the most prevalent colors that a CNN relies on to distinguish a class.

3 IML framework for imbalanced learning

In this section, we outline our framework for applying IML to complex, imbalanced data. Our framework is built on feature embeddings (FE). It starts broadly by visualizing sub-concepts within classes, which we refer to as archetypes. Then, we use nearest adversary classes to gauge error during inference. Next, we visualize class overlap. Finally, our framework allows for zooming in on specific classes to view the most salient colors that define a class. The basic components of the framework are graphically shown in Fig. 1. The components of our framework can be flexibly applied sequentially and in their entirety; or individually, making it fully flexible to the user’s needs. Each component is discussed below.

3.1 Feature embeddings

To make our analysis of imbalanced data complexity more tractable, we work with the low dimensional feature embeddings learned by a CNN. We select the latent representations in the final convolutional layer of a CNN, after pooling. We refer to these features as *feature embeddings (FE)*. FE can be extracted from a trained CNN and used to analyze dataset complexity and to better understand how the model acts on data. FE drawn from the final layer of CNNs capture the central variance in data (Bengio et al., 2013). In computer vision, it has similarly been hypothesized that high dimensional image data can be expressed in a more compact form, based on latent features (Brahma et al., 2015). We use these features, instead of prediction confidence because neural networks can lack calibration and display high confidence in false predictions (Guo et al., 2017).

3.2 Class archetypes

We divide each class into four sub-categories or archetypes: safe, border, rare, and outliers. The archetypes are inspired by Napierala and Stefanowski (2016). The four categories are determined based on their local neighborhood. We use K-nearest neighbors (KNN) to calculate instance similarity with FE instead of input features.

More broadly, the four archetypes facilitate model, dataset and class complexity understanding. We use $K = 5$ to determine the local neighborhood. Our selection of 5 neighbors



Fig. 1 Outline of the main components of our IML framework for imbalanced learning

is consistent with imbalanced resampling methods such as SMOTE (Chawla et al., 2002) and its many variants, as well as Napierala and Stefanowski (2016). Using fewer than 5 neighbors would be challenging with 4 archetypes. More neighbors can be used; however, it may prove difficult with minority classes that have few instances (e.g., in our experiments, the minority class in CIFAR-10 only has 50 instances). The “safe” category represents class instances whose nearest neighbors are from the same class ($N_c = 4$ or $N_c = 5$), where N_c is the number of same class neighbors. Therefore, they are likely homogeneous. The border category are instances that have both same and adversary class nearest neighbors (same class neighbors where $N_c = 2$ or $N_c = 3$) and likely reside at the class decision boundary. The rare category represents class sub-concepts (same class neighbors where $N_c = 1$). Finally, the outlier category are instances that do not have any same class neighbors ($N_c = 0$). In the case of the majority class, outliers may indeed represent noisy instances, whereas for the minority class, the model may classify more instances as outliers due to a reduced number of training examples and the model’s inability to disentangle their latent representations from adversary classes. The four archetypes can be used to select prototypes that can be visualized and further inspected (see Sect. 4.2).

3.3 Nearest adversaries to visualize false positives by class

We believe that the local neighborhood of training instances determined in latent space contains important information about class similarity and overlap. During training, if a CNN embeds two classes in close proximity in latent space, then the model will likely have difficulty disentangling its representations of the two classes during inference (Dablain et al., 2023, 2023). This failure to properly separate the classes during training will likely lead to false positives at validation and test time. Based on this insight, we extract feature embeddings (FE) and their labels from a trained model and use the KNN algorithm to find the K-nearest neighbors of each training instance. If an instance produces a false positive during training, we collect and aggregate the number of nearest adversary class neighbors for each reference class. See Algorithm 1.

In Sect. 4.3, we show visualizations of this technique and how it correlates with validation set false positives using the Kullback Leibler Divergence. In addition, we compare our method to another method that has been used in imbalanced learning to measure class overlap - Fisher’s Discriminant Ratio (FDR):

Extract feature embeddings (FE) for each training set instance.

Run K-Nearest Neighbors on FE.

```

for each instance (i) in Train do
  Determine the class of each nearest neighbor.
  Label each neighbor based on class.
  if the instance is a False Positive then
    Count the number of Adversary Class ( $A_c$ ) neighbors
    Store counts

```

Standardize the counts by Reference Class (R_c) .
 Visualize the counts for each R_c .

Algorithm 1: Nearest Adversaries

$$FDR = \frac{(\mu_{FE_i} - \mu_{FE_k})^2}{\sigma_{FE_i}^2 + \sigma_{FE_k}^2} \quad (1)$$

In Eq. (1), i and k represent pair-wise classes in a dataset and FE is a vector of feature embeddings, where the mean squared difference of latent features (FE) is divided by their variance. As used in Barella et al. (2021), FDR is a measure of how close two classes are, with lower values indicating greater similarity (feature overlap). Thus, like our nearest adversary technique, it can be used to determine class overlap.

3.4 Identify specific class feature map overlap

In the previous section, we examined class overlap at the *instance-level*. Here, we focus on overlapping class *latent features*. Each feature embedding (FE) represents the scalar value of a convolutional feature map (FM), after pooling. These FE/FM are naturally indexed and can be extracted in vector form. This natural indexing allows us to identify the FE's with the highest magnitudes across an entire class.

For each class, the FE magnitudes can be aggregated and averaged. Then, the FE with the largest magnitudes can be selected (the top-K FE). If two classes place a high magnitude on a FE/FM with the same index position, then this FE is important for both classes and hence, may indicate feature overlap. See Algorithm 2. In Sect. 4.4, we provide visualizations of this method, along with suggestions for how it can be used by model users and developers.

3.5 Colors that define classes

Existing IML methods that trace CNN decisions to pixel space via gradient techniques track salient pixel *locations* for *single* image instances. They display a virtual black and white source image (black to indicate high pixel saliency to a CNN's prediction and white to indicate low saliency). We make use of a gradient saliency technique commonly used in IML, which was developed by Simonyan et al. (2013). However, we modify it to trace a prediction to a pixel location only so that we can extract the RGB pixel values at that location.

Extract feature embeddings (FE) for each training set instance.

for each class (c) in Train **do**
 | Calculate the mean of each feature embedding
 | Select the top K FE (e.g., K=10) based on mean
Visualize.

Algorithm 2: Identify Specific Class Feature Map Overlap

Select a Reference Class (R_C).

```

for each instance ( $i$ ) in  $R_C$  do
  Identify Top-K input pixels by back propagating the gradient
  Group pixels into bins based on color bands
  Adaptively update bin means
Visualize.

```

Algorithm 3: Colors that Define Classes

We collect the top-K RGB pixel values for each instance in a training set and also the instance labels. For purposes of our illustrations in Sect. 4.5, we select the top 10% most salient pixels. We partition the collected pixels into bins based on the color spectrum (e.g., black, orange, red, green, blue, light blue, white, etc.). See Algorithm 3.

4 Experiments and results

4.1 Experimental set-up

To illustrate the application of our framework, we select five image datasets: CIFAR-10, CIFAR-100 (Krizhevsky, 2009), Places-10, Places-100 (Zhou et al., 2017) and INaturalist (Van Horn et al., 2018). For each dataset, we use different types and levels of imbalance to highlight varied applications of our framework (see Table 1 for dataset details). For purposes of our experiments, we consider three types of imbalance: exponential (exp.), step and natural. Exponential imbalance is introduced on a gradual basis in a multi-class setting, step imbalance has a cliff effect on the number of instances between classes and natural imbalance depends on data collection (which is unique to the INaturalist dataset).

To make training tractable, we limit Places to 10 and 100 classes and INaturalist to its 13 super-categories. CIFAR-10 and CIFAR-100 are trained with a Resnet-32 (He et al., 2016) backbone and Places and INaturalist with a Resnet-56. Although a Resnet architecture is used for our experiments, any CNN architecture that imposes dimensionality reduction should work (e.g., a (Huang et al., 2017) likely would not facilitate the use of lower

Table 1 Datasets and training

Dataset	Classes	Train	Test	Input dim (pixels)	FE dim	Imbal. type	Max imbal. level	Epochs	Arch.
CIFAR-10	10	12,046	10,000	3072	64	Exp	100:1	200	Res-32
CIFAR-100	100	19,573	5000	3072	64	Exp	10:1	200	Res-32
INaturalist	13	72,358	14,020	1,440,000	64	Natural	50:1	50	Res-56
Places-10	10	15,000	5000	196,608	64	Step	5:1	90	Res-56
Places-100	100	98,072	15,000	196,608	64	Exp	10:1	50	Res-56

dimensional feature embeddings). We adopt a training regime employed by Cao et al. (2019). Except where noted, all models are trained on cross-entropy loss with a single NVIDIA 3070 GPU. As discussed in the following sections, in several cases, we train models with a cost-sensitive method (LDAM) (Cao et al., 2019) to show how visualizations of both baseline and cost-sensitive algorithms can be used for comparative purposes to assess specific areas of improvement.

4.2 Class archetypes

Figure 2 shows the percentage of true positives (TP) for each class in 5 training datasets. The TPs are grouped based on class archetypes: safe, border, rare and outliers. For all of the datasets, the safe and border groups contain the greatest percentage of TPs relative to the total number of instances in the group.

In Fig. 3, we select a prototypical instance from the safe, border, rare and outlier categories for the majority class (airplanes) and the minority class (trucks) from CIFAR-10 for visualization purposes. In large image datasets, it may not be obvious which examples are representative of the overall class (safe examples), which examples reside on the decision boundary (border), and which instances may be sub-concepts or outliers. For each of these

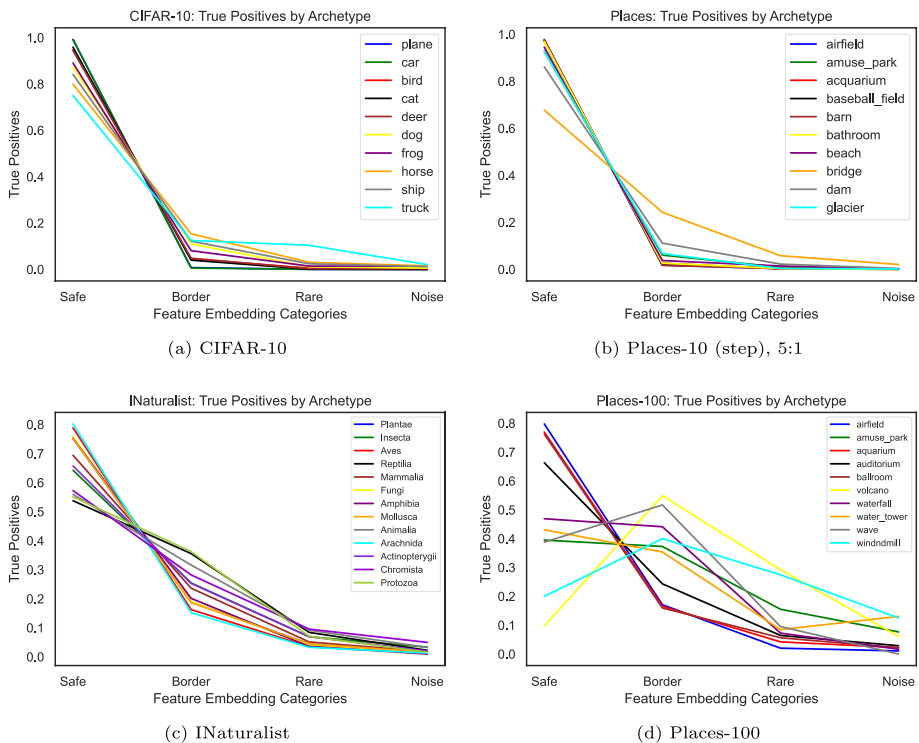


Fig. 2 This figure shows the percentage of True Positives (TPs) of the safe, border, rare and outlier archetypes in each training set. For Places-100, the classes with the 5 largest number of examples and the 5 fewest are shown to make the visualization interpretable. In the sub-figure legends, the class with the greatest number of instances is at the top (majority) and the class with the fewest instances is at the bottom (minority)

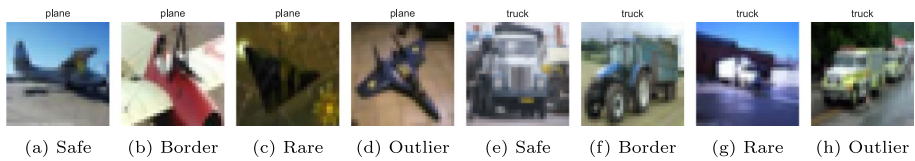


Fig. 3 This figure displays the safe, border, rare and outlier prototypes for 2 classes in the CIFAR-10 dataset. **a–d** are from the class with the largest number of examples (airplanes) and **e–h** are from the class with the fewest number of examples (trucks)

categories and classes, we select the most central prototype, using the K-medoid algorithm, and visualize them.

These visualizations can help identify potential issues that require further investigation of specific classes. For example, in the case of airplanes, it may not be immediately apparent to a human, who preferences shape over texture, why the outlier example is different from the safe prototype in Fig. 3. This seeming incongruence can serve as a flag for model users and algorithm developers. As discussed in more detail in Sect. 4.5, we conjecture that this seeming incongruity is due to a CNN’s preference of texture over shape when distinguishing classes (i.e., there is no blue sky in the outlier airplane prototype).

Use cases For model users, the four categories facilitate the visualization of representative sub-groups within specific classes. When dealing with large datasets, these visualizations can help reduce the need for culling through copious amounts of examples and instead allow model users to focus on a few representative examples: those that are relatively homogeneous in model latent space (safe), those that reside on the decision boundary (border), rare and outlier instances. See Fig. 3.

For imbalanced model developers, the class archetypes can help improve the training process. First, majority class outliers could possibly be mislabeled instances that should be removed. In this case, it may be necessary to examine all of the outlier examples, instead of only the prototype. Second, it can inform potential resampling strategies. For example, safe examples, due to their homogeneity, may be ripe for under-sampling; border and rare instances may be good candidates for over-sampling.

4.3 Nearest adversaries to visualize false positives by class

Figure 4 visualizes the relationship between validation error and training nearest adversaries by class for a CNN trained with the INaturalist dataset. In the figure, each class is represented with a single bar. The class names are matched with specific colors in the legend. In the figure on the left (a), each color in each bar stands for an adversary class that the model falsely predicts as the reference class. The length of the color bars represent the percentage of total false positives produced by the adversary class. In Fig. 4a, we show the validation set false positives.

In contrast, Fig. 4b shows the percentage of adversary class nearest neighbors for each reference class. By placing these diagrams side by side, we can easily compare how nearest adversaries (on the training set) neatly reproduces the classes that will trigger false positives (in the validation set).

This tool can provide a powerful indication of the classes that a model will struggle with during inference when only using training data. Model users and developers can employ the figure on the right as a proxy for the diagram on the left. For example, the training

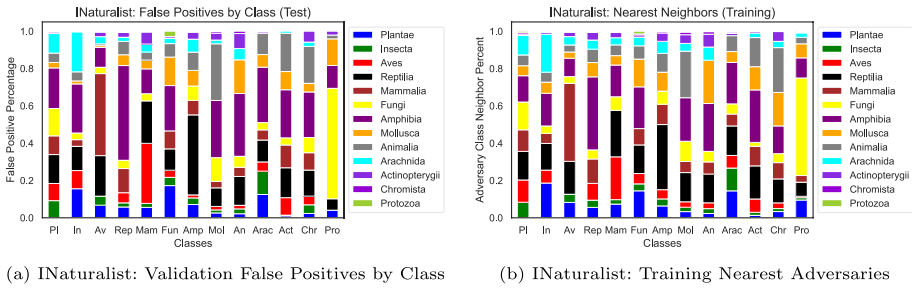


Fig. 4 This figure visualizes the relationship between validation error by class and training nearest adversaries by class. This tool can provide a powerful indication of the classes that a model will struggle with during inference

nearest adversary neighbors diagram (b) quickly shows, and the validation set diagram (a) confirms, that the model has the most difficulty (FPs) with: Fungi for the Protozoa class, Mammals for the Aves (birds) class, Amphibia (fish) for the reptile class, and reptiles for the Amphibia class.

In order to confirm the ability of training set nearest adversaries to predict the classes that a model will produce more false positives during validation, we measure the difference in the nearest adversary and validation FP distributions. We use Kullback Liebler Divergence (KLD) (Kullback & Leibler, 1951) to measure the difference in these distributions for five datasets. We also compare our nearest adversary prediction with two other methods: a random distribution and Fisher’s Discriminant Ratio. Table 2 shows that our method (NNB) predicts much better than random (by a factor between 1.8 and 34 times better) and compares favorably with another method that measures class overlap, Fisher’s Discriminant Ratio (see Sect. 3.3 for a description of FDR). Although FDR may be more accurate in some cases, it only shows pairwise similarity of classes. In contrast, our NNB method visualizes the proportionate similarity of all adversary classes compared to a reference class so that a tiered spectrum of overlap for classes in a dataset can be readily seen, offering a more realistic outlook on the difficulty of the considered dataset.

This simple tool is useful because it is an indicator of latent feature entanglement. If a model produces a large amount of adversary instances that are in close proximity in the training set to the reference class, then the model will likely have difficulty distinguishing the two close neighbors at validation time.

Table 2 KLD of Validation Set False Positives

Dataset	NNB KLD	FDR KLD	Rand. KLD	NNB: Rand. Factor
CIFAR-10	0.5561	0.5157	4.535	8.155
CIFAR-100	4.039	1.577	7.437	1.841
Places-10	0.4340	0.2942	2.821	6.500
Places-100	0.5919	1.327	4.191	7.081
INaturalist	0.0577	.3537	1.988	34.47

The bolded values indicate the top-performing method

Use cases This technique can be a powerful tool for imbalanced data. Our method allows model users and imbalanced algorithm developers to gauge the classes that the model will have difficulty with. Therefore, our visualization allows users to reasonably predict the distribution of validation error based solely on the training set.

4.4 Feature map overlap

In the previous section, we visualized class overlap at the instance level. Here, we examine class overlap at the feature embedding (FE) level. FE are scalar values of the output of a CNN’s final convolutional layer, after pooling. Higher valued FE indicate CNN feature maps, in the last convolutional layer, that the model views as more important for object classification purposes.

In Fig. 5, we visualize the ten most significant FE for each class in CIFAR-10 (i.e., the ones with the largest mean magnitudes for each class). Each bar represents a class, as shown on the *x*-axis. Each of the 10 segments of each bar is color coded, such that gray is the FE with the largest mean magnitude (on the bottom of the bar) and pink is the smallest (top of the bar). Each color coded segment of a bar contains a number, which is the index of a FE/FM. For this model, there are 64 FE/FM. The relative size of each segment (*y*-axis) shows the percentage that each FE magnitude makes up of the top-10 FE magnitudes.

Therefore, the chart shows the most important latent features (feature maps) that a CNN uses for each class to make its class decision. Because the FE indices are shown for each class, they can be compared between classes to identify latent feature overlap.

For example, in Fig. 5a, we can see that trucks and cars contain five common FE in their top-10 most important FE (i.e., FE indices 57, 53, 43, 0 and 44). In contrast, trucks and planes share only 2 top-10 FE (FE indices 43 and 53). Trucks are the class with the fewest number of training examples, with planes the most, and cars the next largest. For trucks, the two classes that produce the most false positives at validation time are cars and planes, respectively (see Fig. 6a). This chart implies that the large number of FPs produced for planes and cars may be due to two different factors. In the case of planes, it seems to be due

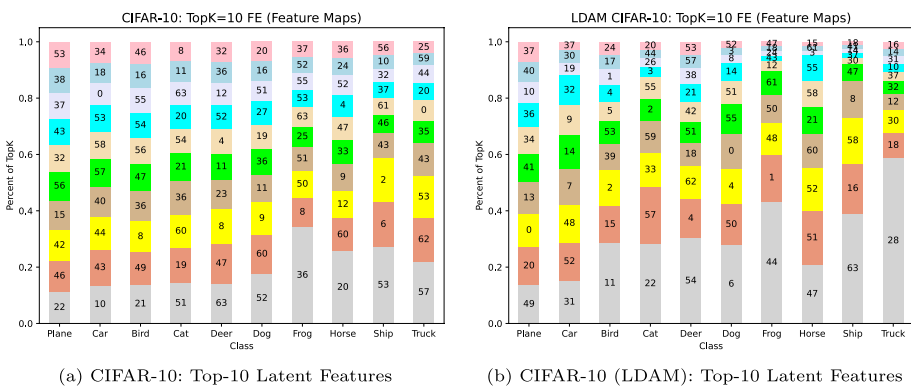


Fig. 5 This diagram provides a clear indication of class overlap at the feature map level. It shows the top-K ($K = 10$) latent features (FE) used by the model to predict CIFAR-10 classes. Each of the 10 segments of each bar is color coded, such that gray is the FE with the largest mean magnitude (on the bottom of the bar) and pink is the smallest (top of the bar). In this case, there are a total of 64 feature maps, which correspond to the FE index numbers listed in the bar charts. Each number in the bar chart represents a FE or feature map index

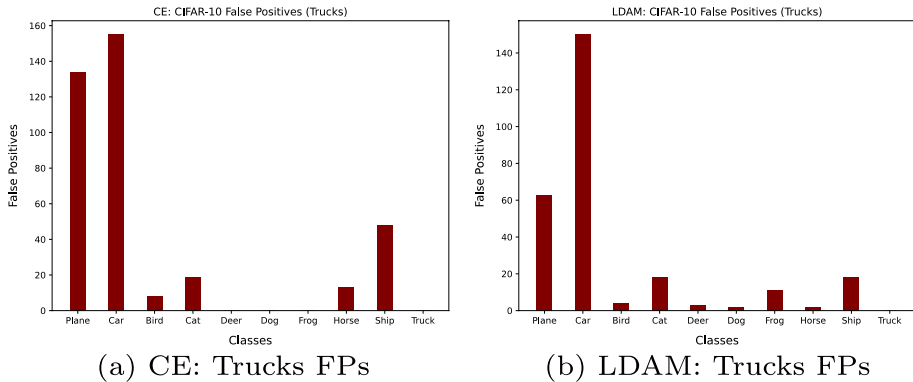


Fig. 6 This diagram shows the false positives for trucks for CNNs trained with cross-entropy loss (CE) and LDAM

to numerical differences in the number of training examples because of the low FE overlap; whereas, in the case of cars, it appears to be due to FE overlap.

We can further explore this hypothesis by examining how a cost-sensitive algorithm, LDAM, which focuses on the numerical difference of training instances (and not features) behaves in the face of class overlap. In Fig. 5, the figure on the left (a) is trained with cross-entropy loss and the figure on the right (b) is trained with a popular cost-sensitive method used in imbalanced learning, LDAM. Interestingly, in the figure on the right (b), where a CNN is trained with a cost-sensitive method, there is still five FE that are shared in common between the truck and car classes. In fact, if we view figures (a) and (b) of Fig. 6, we can see that LDAM reduces false positives for the plane class but does not have a large impact on the automobile class, which is likely because it is geared toward addressing *instance numerical differences* and *not latent feature overlap*. Thus, although the cost-sensitive method may have addressed the class imbalance, in part, it does not appear to have completely addressed feature overlap.

Use cases This visualization can provide vital clues about where a CNN classifier may break-down. The cause of FPs may not always be solely due to class imbalance. Other factors, such as a model’s entanglement of latent features, may be at stake. In these situations, imbalanced learning algorithm developers may want to consider techniques that address feature entanglement, instead of solely class numerical imbalance. For example, it may be possible to design cost-sensitive loss functions that assign a greater cost to FE overlap based on FE index commonality between classes.

This visualization may also be used to assess cost-sensitive algorithms. The visualization can help imbalanced learning algorithm developers decide if, for example, cost-sensitive techniques are addressing only class imbalance or, additionally, if their methods improve feature entanglement in latent space [see also (Ghosh et al., 2022; Pazzani et al., 1994)].

4.5 Colors that define classes

This visualization can be used to identify the color bands that are most prevalent in a data class. As an illustration, Fig. 7 shows the top 10% of color group textures for the truck, auto and plane classes in CIFAR-10. In the case of autos and trucks, black

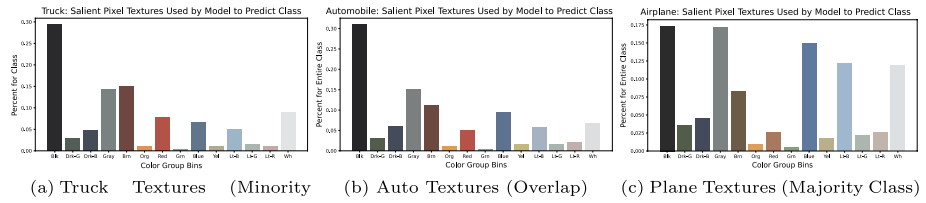


Fig. 7 This diagram shows the top 10% of color groups for specific classes based on gradient saliency tracing. The classes are drawn from CIFAR-10

(30%) and gray (15%) are the 2 most common colors. Since all cars and trucks have (black) tires, the presence of this color is not surprising. Even though the number of samples is vastly different between cars and trucks (60:1 imbalance), the overall proportion of color bands is very similar, which tracks the FE space overlap that we previously observed for these 2 classes (model feature entanglement). In the case of planes, black and gray are still important (17% each); however, there is a much larger percentage of blue, light blue, and white (12.5% each), due to the greater presence of blue sky and white clouds (background). In contrast, white is salient only 5% of the time for cars and trucks.

Additionally, Fig. 8 shows safe and border prototypes for a baseball field (majority class) and rafts (minority class) from a CNN trained on the Places-100 dataset with

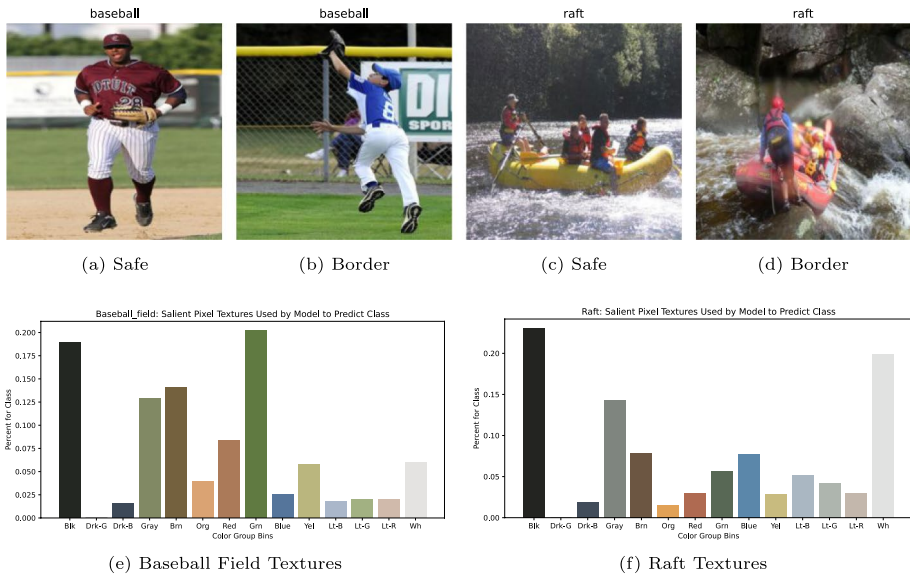


Fig. 8 This diagram shows the most salient colors for a majority (baseball field) and minority class (raft), along with archetypical images drawn from the safe and border categories and a CNN trained on the Places-100 dataset. In the case of a baseball field, the model preferences green, brown and gray; whereas for rafts, white is more prevalent (likely due to white rapids) and brown and green are less emphasized. This type of information may be relevant for purposes of oversampling techniques in pixel space. By determining the colors that the model preferences, it may be possible to modify the colors via augmentation to train the model to preference other colors (Color figure online)

cross-entropy loss. For the baseball field, the most salient colors used by the model to detect class instances are black, green and brown. In the safe prototype, we can see black leggings on the player's uniform, green grass and a brown infield. In the border prototype, we can see a black background (over the fence), player black shoes, and green grass. In the case of rafts, green and brown are not as prevalent in the model's top 10% most salient pixels. Instead, white (white water rapids) and the black background are more important.

Use cases Users of CNNs trained on imbalanced data may use this visualization to better understand the major color bands that are prevalent across a class. When combined with class prototype visualization, it can also provide intuition into whether a classifier is using background colors (e.g., blue sky or clouds) to discern a class. For imbalanced learning algorithm developers, it can suggest specific pixel color groups that may be over- or under-sampled at the front-end of image processing to improve classifier accuracy.

5 Limitations and future directions

There are several potential limitations to our research that should be seen as future directions in developing XAI and IML systems for imbalanced data. First, our techniques were applied to datasets comprising object recognition in natural scenes. A future research direction could be to extend these techniques to object detection and in-door settings. Second, we focused on datasets where the number of class instances were imbalanced. A potential future research direction could be to extend our research to adversarial example analysis. For example, when an adversarial instance is misclassified, (1) which feature embeddings caused the misclassification, (2) what is the distribution of classes that are falsely predicted by adversarial examples, and (3) which input image colors or features does the model struggle when small perturbations are made to an image class?

6 Conclusion

We present a framework that can be used by both model users and algorithm developers to better understand and improve CNNs that are trained with imbalanced data. Because modern neural networks depend on large quantities of data to achieve high accuracy, understanding how these models use complex data and are affected by class imbalance is critical. Our Class Archetypes allow model users to quickly identify a few prototypical instances in large datasets for visual inspection to determine safe, border, rare and outlier instances of a class in datasets with multiple classes and a large number of examples. Our Nearest Adversaries visualization enables model users and developers to identify specific classes that overlap in a multi-class setting and provides a “heatmap” of the classes causing the greatest overlap. Our feature overlap visualization allows model users to identify specific latent features that are overlap and cause model confusion. Finally, our colors that define classes technique permits model users to understand specific colors that a model relies in making decisions for an entire class which provides insight into potentially spurious feature selection and the role of background scene context on model decisions.

Author Contributions All authors read and wrote the paper.

Funding No funding.

Data availability statement Data is publicly available.

Code availability Code is publicly available.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval Not applicable.

Consent to participate All authors consent to participate.

Consent for publication All authors consent to publication.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References


- Achtibat, R., Dreyer, M., Eisenbraun, I., Bosse, S., Wiegand, T., Samek, W., & Lapuschkin, S. (2022). From “where” to “what”: Towards human-understandable explanations through concept relevance propagation. arXiv preprint [arXiv:2206.03208](https://arxiv.org/abs/2206.03208).
- Aha, D. W. (1992). Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. *International Journal of Man-Machine Studies*, 36(2), 267–287.
- Artelt, A., & Hammer, B. (2019). On the computation of counterfactual explanations—A survey. arXiv preprint [arXiv:1911.07749](https://arxiv.org/abs/1911.07749).
- Artelt, A., & Hammer, B. (2020). Convex density constraints for computing plausible counterfactual explanations. In *Artificial neural networks and machine learning—ICANN 2020: 29th international conference on artificial neural networks, Bratislava, Slovakia, September 15–18, 2020, Proceedings, Part I 29* (pp. 353–365). Springer.
- Baker, N., Lu, H., Erlikhman, G., & Kellman, P. J. (2018). Deep convolutional networks do not classify based on global object shape. *PLoS Computational Biology*, 14(12), 1006613.
- Barella, V. H., Garcia, L. P., Souto, M. C., Lorena, A. C., & Carvalho, A. C. (2021). Assessing the data complexity of imbalanced datasets. *Information Sciences*, 553, 83–109.
- Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1), 20–29.
- Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798–1828.
- Bien, J., & Tibshirani, R. (2011). Prototype selection for interpretable classification. *The Annals of Applied Statistics*, 5(4), 2403–2424.
- Brahma, P. P., Wu, D., & She, Y. (2015). Why deep learning works: A manifold disentanglement perspective. *IEEE Transactions on Neural Networks and Learning Systems*, 27(10), 1997–2008.
- Bruckert, S., Finzel, B., & Schmid, U. (2020). The next generation of medical decision support: A roadmap toward transparent expert companions. *Frontiers in Artificial Intelligence*, 3, 507973.
- Burkart, N., & Huber, M. F. (2021). A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70, 245–317.
- Cao, K., Wei, C., Gaidon, A., Aréchiga, N., & Ma, T. (2019). Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in neural information processing systems 32: annual conference on neural information processing systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada* (pp. 1565–1576).
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357.

- Chen, C., Lin, K., Rudin, C., Shaposhnik, Y., Wang, S., & Wang, T. (2018). An interpretable model with globally consistent explanations for credit risk. arXiv preprint [arXiv:1811.12615](https://arxiv.org/abs/1811.12615).
- Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), 21–27.
- Dablain, D., Bellinger, C., Krawczyk, B., Chawla, N. (2023). Efficient augmentation for imbalanced deep learning. In *IEEE 39th international conference on data engineering*.
- Dablain, D., Jacobson, K.N., Bellinger, C., Roberts, M., & Chawla, N. (2023). Understanding CNN fragility when learning with imbalanced data. *Machine Learning*, 1–26.
- Denil, M., & Trappenberg, T. (2010). Overlap versus imbalance. In *Canadian conference on artificial intelligence* (pp. 220–231). Springer.
- Fix, E., & Hodges, J. L. (1989). Discriminatory analysis. Nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3), 238–247.
- García, V., Sánchez, J., & Mollineda, R. (2007). An empirical study of the behavior of classifiers on imbalanced and overlapped data sets. In *Iberoamerican congress on pattern recognition* (pp. 397–406). Springer.
- Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., Brendel, W. (2018). Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint [arXiv:1811.12231](https://arxiv.org/abs/1811.12231)
- Ghosh, S., Baranowski, E. S., Biehl, M., Arlt, W., Tino, P., & Bunte, K. (2022). Interpretable models capable of handling systematic missingness in imbalanced classes and heterogeneous datasets. arXiv preprint [arXiv:2206.02056](https://arxiv.org/abs/2206.02056).
- Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th international conference on data science and advanced analytics (DSAA)* (pp. 80–89). IEEE.
- Gunning, D., & Aha, D. (2019). Darpa’s explainable artificial intelligence (XAI) program. *AI Magazine*, 40(2), 44–58.
- Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International conference on machine learning* (pp. 1321–1330). PMLR.
- Haralick, R. M. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5), 786–804.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).
- Hermann, K., Chen, T., & Kornblith, S. (2020). The origins and prevalence of texture bias in convolutional neural networks. *Advances in Neural Information Processing Systems*, 33, 19000–19015.
- Hoffman, R. R., Mueller, S. T., Klein, G., & Litman, J. (2018). Metrics for explainable ai: Challenges and prospects. arXiv preprint [arXiv:1812.04608](https://arxiv.org/abs/1812.04608).
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708).
- Huber, T., Weitz, K., André, E., & Amir, O. (2021). Local and global explanations of agent behavior: Integrating strategy summaries with saliency maps. *Artificial Intelligence*, 301, 103571.
- Jo, T., & Japkowicz, N. (2004). Class imbalances versus small disjuncts. *ACM Sigkdd Explorations Newsletter*, 6(1), 40–49.
- Kabra, M., Robie, A., & Branson, K. (2015). Understanding classifier errors by examining influential neighbors. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3917–3925).
- Keane, M. T., & Kenny, E. M. (2019). How case-based reasoning explains neural networks: A theoretical analysis of xai using post-hoc explanation-by-example from a survey of ann-cbr twin-systems. In *International conference on case-based reasoning* (pp. 155–171). Springer.
- Kenny, E. M., Ford, C., Quinn, M., & Keane, M. T. (2021). Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in xai user studies. *Artificial Intelligence*, 294, 103459.
- Krizhevsky, A. (2009). *Learning multiple layers of features from tiny images*. Master’s thesis, University of Toronto.
- Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1), 79–86.
- Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., Kolter, J.Z., Langer, D., Pink, O., & Pratt, V. (2011). Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE intelligent vehicles symposium (IV)* (pp. 163–168). IEEE.

- Linaratos, P., Papastefanopoulos, V., & Kotsiantis, S. (2020). Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1), 18.
- Lipton, Z. C. (2018). The myths of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3), 31–57.
- Marcus, G. (2018). Deep learning: A critical appraisal. arXiv preprint [arXiv:1801.00631](https://arxiv.org/abs/1801.00631).
- McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint [arXiv:1802.03426](https://arxiv.org/abs/1802.03426).
- Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267, 1–38.
- Mothilal, R. K., Sharma, A., & Tan, C. (2020). Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency* (pp. 607–617).
- Napierala, K., & Stefanowski, J. (2016). Types of minority class examples and their influence on learning classifiers from imbalanced data. *Journal of Intelligent Information Systems*, 46(3), 563–597.
- Papernot, N., & McDaniel, P. (2018). Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. arXiv preprint [arXiv:1803.04765](https://arxiv.org/abs/1803.04765).
- Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., & Brunk, C. (1994). Reducing misclassification costs. In *Machine learning proceedings* (pp. 217–225). Elsevier.
- Prati, R. C., Batista, G. E., & Monard, M. C. (2004). Class imbalances versus class overlapping: An analysis of a learning system behavior. In *Mexican international conference on artificial intelligence* (pp. 312–321). Springer.
- Ras, G., Xie, N., Gerven, M., & Doran, D. (2022). Explainable deep learning: A field guide for the uninitiated. *Journal of Artificial Intelligence Research*, 73, 329–397.
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135–1144).
- Schulz, A., Hinder, F., & Hammer, B. (2019). Deepview: Visualizing classification boundaries of deep neural networks as scatter plots using discriminative dimensionality reduction. arXiv preprint [arXiv:1909.09154](https://arxiv.org/abs/1909.09154).
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 618–626).
- Shapley, L. S. (1953). *A value for n-person games*. Princeton University Press.
- Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint [arXiv:1312.6034](https://arxiv.org/abs/1312.6034).
- Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. In *International conference on machine learning* (pp. 3319–3328). PMLR.
- Teach, R. L., & Shortliffe, E. H. (1981). An analysis of physician attitudes regarding computer-based clinical consultation systems. *Computers and Biomedical Research*, 14(6), 542–558.
- Tjoa, E., & Guan, C. (2020). A survey on explainable artificial intelligence (XAI): Toward medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11), 4793–4813.
- Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., & Belongie, S. (2018). The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8769–8778).
- Weiss, G. M. (2004). Mining with rarity: A unifying framework. *ACM Sigkdd Explorations Newsletter*, 6(1), 7–19.
- Xie, Y., Pongsakornsathien, N., Gardi, A., & Sabatini, R. (2021). Explanation of machine-learning solutions in air-traffic management. *Aerospace*, 8(8), 224.
- Ye, L. R., Johnson, P. E. (1995). The impact of explanation facilities on user acceptance of expert systems advice. *Mis Quarterly*, 157–172.
- Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. In *International conference on learning representations*.
- Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2921–2929).
- Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., & Torralba, A. (2017). Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6), 1452–1464.
- Zilke, J. R., Loza Mencía, E., & Janssen, F. (2016). Deepred—rule extraction from deep neural networks. In *International conference on discovery science* (pp. 457–473). Springer.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Damien Dablain¹  · Colin Bellinger² · Bartosz Krawczyk³ · David W. Aha⁴ · Nitesh Chawla¹

✉ Nitesh Chawla
nchawla@nd.edu

Damien Dablain
ddablain@nd.edu

Colin Bellinger
colin.bellinger@nrc-cnrc.gc.ca

Bartosz Krawczyk
bkrawczyk@vcu.edu

David W. Aha
david.aha@nrl.navy.mil

¹ Lucy Family Institute for Data and Society, Department of Computer Science, University of Notre Dame, Notre Dame, IN 46556, USA

² National Research Council of Canada, Ottawa K1A, Canada

³ Dept. Computer Science, Virginia Commonwealth University, Richmond, VA 23824, USA

⁴ Navy Center of Applied Research on AI, Naval Research Laboratory, Washington, DC 20375, USA