A Machine Teaching Framework for Scalable Recognition

Pei Wang
UC, San Diego
pew062@ucsd.edu

Nuno Vasconcelos
UC, San Diego
nuno@ucsd.edu

Abstract

We consider the scalable recognition problem in the fine-grained expert domain where large-scale data collection is easy whereas annotation is difficult. Existing solutions are typically based on semi-supervised or self-supervised learning. We propose an alternative new framework, MEMORABLE, based on machine teaching and online crowdsourcing platforms. A small amount of data is first labeled by experts and then used to teach online annotators for the classes of interest, who finally label the entire dataset. Preliminary studies show that the accuracy of classifiers trained on the final dataset is a function of the accuracy of the student annotators. A new machine teaching algorithm, CMaxGrad, is then proposed to enhance this accuracy by introducing explanations in a state-of-the-art machine teaching algorithm. For this, CMaxGrad leverages counterfactual explanations, which take into account student predictions, thereby proving feedback that is student-specific, explicitly addresses the causes of student confusion, and adapts to the level of competence of the student. Experiments show that both MEMORABLE and CMaxGrad outperform existing solutions to their respective problems.

1. Introduction

The success of deep learning in computer vision has been largely driven by large-scale datasets. Many breakthroughs, made across various tasks, have benefited from large-scale and well-curated datasets like ImageNet for object recognition [6], COCO for object detection and segmentation [23], Kinetics for action recognition [19], etc. These datasets usually contain common objects, scenes, or actions and thus can be scalably annotated on crowdsourcing platforms such as Amazon Mechanical Turk (MTurk) [14]. When this is possible, we say that *learning is scalable.* However, this is usually not the case for expert domains, such as biology or medical imaging. While data collection can still be easy in these domains, annotations require highly specialized and domain-specific knowledge. For example, while it is easy to crawl the web or deploy cameras in the wild to collect a large number of animal images, it is usually expensive to recruit the biologists or taxonomists needed to label them. The resulting lack of large annotated datasets hampers the application of deep learning to expert domains. For example, the largest existing bird dataset, NAbirds, only contains about 48k instances [37]. Even the recent and largest biological dataset, iNaturalist, contains only about 850k instances [38]. This is smaller than ImageNet, proposed about 10 years ago, and pales in comparison to the largest datasets of everyday objects, e.g. Open Images with 9M images [20].

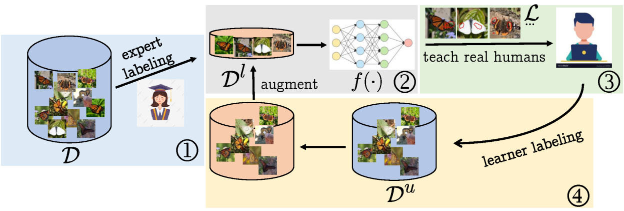


Figure 1: The proposed MEMORABLE framework is a new solution to the problem of large-scale recognition in fine-grained domains. ① A large-scale raw dataset ($\mathcal{D}$) is collected and a small subset delivered to experts, who produce a labeled dataset $\mathcal{D}^l$; ② A neural network is trained on $\mathcal{D}^l$ or semi-supervised trained on $\mathcal{D}^l \cup \mathcal{D}^u$, where $\mathcal{D}^u = \mathcal{D} - \mathcal{D}^l$; ③ A teaching tutorial, composed of a teaching set $\mathcal{L}$ of images and associated explanations, is created from $\mathcal{D}^l$ with the CMaxGrad algorithm, and used to teach human annotators the target categories; ④ human annotators label the unlabeled data $\mathcal{D}^u$. Finally, the classifier is re-trained on the fully labeled datset $\mathcal{D}$. (Red and blue cylinders represent whether data is labeled or not.)

Since labeling is difficult in expert fine-grained domains, scalable learning must take advantage of small expert-labeled datasets and large amounts of unlabeled data. This motivated extensive research on less label-intensive forms of learning, including few-shot learning, transfer learning, semi-supervised learning, and self-supervised learning. For example, models pre-trained on an everyday domain by supervised learning are frequently transferred to a target fine-grained domain by fine-tuning. Another strategy is to learn a good feature extractor by self-supervised learning, which requires no labels, and then fine-tune a classifier at the top of it on a small set of labeled target data. However, these approaches usually underperform scalable supervised learning. For example, state-of-the-art self-supervised learning with SimCLR [3] underperforms a supervised baseline when only a subset of the samples are labeled, especially on fine-grained domains [43].

Unlike all these approaches, we pursue the alternative solution of scaling up the process of *data annotation.* While this was a pie in the sky idea in the past, two recent developments now make it promising. First, several crowdsourcing platforms, like Amazon Mechanical Turk, Sama [15], microWorkers [13], or Clickworker [12], have appeared in recent years, making it easier to recruit large numbers of image annotators online. Second, research has been steadily increasing in the area of machine teaching [48, 47, 28], showing potential to develop algorithms capable of teaching these annotators the domain-specific knowledge needed to label expert data. While these developments are promising, there have been so far no efforts to study how they can be combined into a complete framework for scalable learning. Typically, machine teaching papers only evaluate the accuracy of the labeling produced by the annotators taught by their algorithms. While this is informative, it does not fully address the scalable learning problem, which also includes the design of deep learning systems using those annotations. This raises an additional set of questions, such as what quality must the labels have to guarantee effective deep learning performance, how can the machine teaching algorithms achieve that quality, and whether noisy label learning algorithms [22, 9] have a role in the process.

In this work, we address these questions in the context of scalable learning of recognition systems, which we denote as *scalable recognition.* We propose a new *Machine tEaching fraMewORk for scAlaBLe rEcognition* (MEMORABLE) in fine-grained expert domains, illustrated in Figure 1. A large raw dataset ($\mathcal{D}$) is first collected for a target fine-grained task, e.g. by deploying cameras in the wild or crawling archived medical images in a hospital database. A small subset $\mathcal{D}^l \subset \mathcal{D}$ and $|\mathcal{D}^l| \ll |\mathcal{D}|$ is then labeled by experts. Machine teaching is next used to teach non-experts, e.g. Amazon MTurk workers, how to label for the target categories. The unlabeled data $\mathcal{D}^u = \mathcal{D}/\mathcal{D}^l$ is finally labeled by these humans and the complete dataset used to train an image recognition system. To identify critical areas of this framework, we perform an initial study with simulated noisy annotations. This shows that the accuracy of the machine teaching plays a significant role in the accuracy of the final recognition system. We then hypothesize that better machine teaching performance can be achieved by introducing explanations in the machine teaching algorithm. State-of-the-art machine teaching algorithms [25, 26, 18] tend not to use explanations. Although there is literature doing [28], it tends to rely on attributive explanations [33, 46] that do not take into account the student predictions. To address this problem, we propose the addition of counterfactual explanations to machine teaching.

Counterfactual explanations [41, 8] take into account both ground-truth labels and student predictions, highlighting image regions that are most discriminant of student mistakes. They are thus most instructive for humans to learn from their errors. Furthermore, because the explanatory feedback varies according to the student's prediction, they naturally adjust to the level of competence of the student. We seek to leverage all these benefits by introducing a generalization of the recent MaxGrad machine teaching algorithm [40], denoted *Counterfactual MaxGrad* (CMaxGrad), which is endowed with counterfactual explanations. Experiments show that this algorithm both achieves state-of-the-art machine teaching performance and enables significant scalable recognition gains for the MEMORABLE framework. The latter is itself shown to outperform other scalable recognition strategies, such as semi-supervised learning. It is also shown that deep learning systems trained with MEMORABLE can leverage noisy label training schemes with surprising effectiveness.

The contributions of the paper are summarized as 1) a study of the importance of labeling accuracy for the accuracy of scalable recognition; 2) the MEMORABLE framework to solve the fine-grained scalable recognition problem, by leveraging crowdsourcing platforms and machine teaching algorithms; 3) the new CMaxGrad machine teaching algorithm that introduces counterfactual explanations into machine teaching; and 4) new benchmarks, based on two challenging datasets, for the evaluation of scalable recognition.

2. Related Work

Crowdsourcing platforms There are two types of crowd sourcing platforms. They provide expert and non-expert annotation services. Amazon Mechanical Turk [14] is a widely known and representative one. It has been making it easy to require simple annotation tasks of significantly huge size to a large pool of workers. Although Amazon Turk has been broadly used, most of the workers are non-expert for a specific target expertise task like fine-grained annotation. For example, they can help annotate "dog" and "cat", but hard to do "California Gull" and "Western gull". The lack of prior knowledge of a specific domain makes it hard to satisfy the requirement of fine-grained expert domain labeling. The similar platforms include Sama [15], microWorkers [13], Clickworker [12], etc. They all provide similar services just with slight differences. A comprehensive discussion of them can be found in [32].

Another type of crowdsourcing platform can give expertise annotation service. Citizen scientist is a typical one [37]. It is non-profit and people in this platform are nonprofessional scientists or enthusiasts in a particular domain. They contribute annotations with the understanding that their expertise, experience and passion in a domain of interest. Although it makes it feasible to do expert labeling, there are some problems. Because of non-profits, it is hard to guarantee the quality of their results and guarantee that they are all responsible. This is different from Amazon Turk where if the annotation results are assessed badly by the requester, the worker would not get the payment. The second problem is that the active user number is small, especially on some minor domains. So it is hard to meet the large-scale annotation requirement. In this work, we use Amazon Turk, but unlike the common usage, a short course is introduced preceding the annotation. The worker is trained first and then annotates. This alleviates the problems of both types.

Semi-supervised and self-supervised learning Semi-supervised learning describes a class of algorithms that seek to learn from both unlabeled and labeled samples, typically assumed to be sampled from the same or similar distributions. Limited to the space, we refer to [44] for an extensive survey and [45] for up-to-date development.

Self-supervised learning (SSL) refers to learning methods in which the model is explicitly trained with supervisory signals that are generated from the data itself by leveraging some pretext tasks. The pretext tasks can be predictive tasks, generative tasks, contrasting tasks, or a combination of them. SSL can benefit almost all types of downstream tasks, e.g. semi-supervised learning, that can also be used to evaluate the quality of features learned by self-supervised learning [3, 4, 2]. Literature [17, 27, 16] is recommended for an extensive overview.

Counterfactual explanations Given an image of class A and a user-specified counterfactual class B, counterfactual explanations produce an explanation to answer "why the prediction is A but not B" [39, 24, 49, 1, 10]. In computer vision, the explanations are usually given by visualizations. Two main approaches to these explanations have emerged. The first group is based on an image transformation that elicits the classification as B [39, 24, 49]. The simplest example is adversarial attack [7, 39], which optimize perturbations to map an image of class A into class B. However, adversarial perturbations usually push the perturbed image outside the boundaries of the space of natural images. A more plausible alternative is to exhaustively search the space of features extracted from a large collection of images, to find replacement features that map the image from class A to B [8]. However, exhaustive search is too complex for interactive applications. Another form is optimization-free but produces a pair of segments on two images from ground truth class and counterfactual class [41]. These segments cover the class-discriminant regions. Its generation is much faster and we use it in our work.

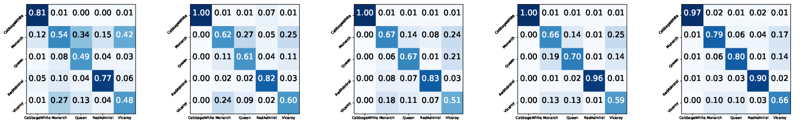
Machine teaching Machine teaching is a broad area. The goal is to select a small number of data from a large set so that this small set can efficiently teach a student. The student can be either a network model or a real human. Because this paper mainly talks about the latter, we recommend [48, 47] for the reader about the network-oriented machine teaching. For real-human machine teaching, a typical strategy is to first model humans as a network model and then select a teaching sequence universally used for human teaching. In this process, most of the previous literature simulates human students based on the assumption that they have limited capacity or are otherwise suboptimal learners [34, 18, 30]. This is intuitive but not optimal in the crowdsourcing context, which has been discussed in [40]. The latter is subject to an optimal student assumption that the students will try their best to complete the assigned tasks. Another direction of real-human machine teaching is to think about how to incorporate the explanation into the teaching process because it is straightforward that explanations are helpful for digesting the knowledge easily [28, 5, 36]. A representative work [28] merges the attribution map into the example selection and feedback stage of teaching. When the learner makes a mistake, a heatmap [46] that highlights the regions that contribute to the correct class is shown. This, to a certain extent, provides some explanations but can not adapt to the learner's choice. Counterfactual explanations were simply associated with random selected images to evaluate their qualities in [41, 8], but there is no special machine teaching algorithm involved and the evaluation is only on simple binary classification tasks. Tropel [31] lets workers identify positive/negative images with respect to a given query image, to train a detector. This is unlike a counterfactual explanation for teaching, where the counter class is an incorrect label chosen by the worker. The latter more directly provides the worker with feedback regarding mistakes. Also, there is no image-based explanation in Tropel. In this paper, we attempt to include the counterfactual explanation into the machine teaching, an explanation that explicitly indicates the class-discriminant between correct class and mis-chosen class. The experiments show that this is more helpful.

3. The MEMORABLE Framework

In this section we introduce the MEMORABLE framework.

3.1. Machine Teaching

We consider the problem of C-class classification on expert domains where data collection is easy but annotation is difficult. For example, while biologists routinely deploy camera traps in the wild [29] or underwater [11], the labeling of the resulting images by professional taxonomists is quite expensive. The goal is to train classifiers



(a) RANDOM [28] (b) omniIMT [25] (c) imiIMT [25] (d) bbIMT [26] (e) MaxGrad [40]

Figure 2: Confusion matrices for human annotators trained by different machine teaching algorithms on Butterflies dataset [28].

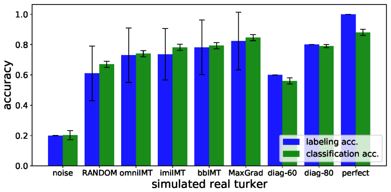


Figure 3: Labeling and classification accuracies of simulated turkers.

from large datasets, i.e. scalable recognition. A practical solution is semi-supervised learning. A large set of images $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{M+N}$ is first collected and a small subset $\mathcal{D}^a = \{\mathbf{x}_i\}_{i=1}^{M}$, where $M \ll N$ labeled by experts. This results in a labeled dataset $\mathcal{D}^l = \{(\mathbf{x}_i, y_i)\}_{i=1}^{M}$ where y_i is the label of $\mathbf{x}_i$. A classifier f is then learned from the semi-supervised dataset $\mathcal{D}^s = \mathcal{D}^u \cup \mathcal{D}^l$, where $\mathcal{D}^u = \mathcal{D} - \mathcal{D}^a$ is the set of unlabeled images. The performance of f is finally evaluated on a testing set $\mathcal{T}$. While various semi-supervised learning algorithms exist [45, 44], their performance is frequently inferior to supervised learning. This gap can be bridged by labeling the data $\mathcal{D}^u$ on a crowd-sourcing platform, such as Amazon MTurk. This, however, is impossible for data from domains, e.g. animal taxonomies, on which MTurk annotators have no expertise.

MEMORABLE addresses this problem by leveraging the labeled dataset $\mathcal{D}^l$ to *teach* MTurk annotators to label the images in $\mathcal{D}^u$. As shown in Figure 1, this is done in several steps. A classifier f is first trained, either by semi-supervised learning on $\mathcal{D}^l \cup \mathcal{D}^u$, or supervised learning on $\mathcal{D}^l$. This classifier is then leveraged to design a *teaching set* $\mathcal{L} \subset \mathcal{D}^l$ of $L \ll M$ images for training MTurk annotators. Several machine teaching algorithms have been proposed to extract an optimal teaching set from $\mathcal{D}^l$ [28, 34, 40]. Finally, MTurk annotators are trained by practicing on the teaching set $\mathcal{L}$. This usually consists of an introductory step where they are shown one (or a few) images of each class, and an iterative step where they attempt to classify images in $\mathcal{L}$ and receive feedback on their mistakes. When this process is completed, the trained MTurkers are finally asked to label $\mathcal{D}^u$ and the classifier is retrained.

While various works have addressed individual components of this framework, e.g. by proposing different machine teaching algorithms [28, 34, 40] or semi-supervised learning techniques [45, 44], we are aware of no studies on the effectiveness of the entire scalable recognition architecture. Two questions, in particular, seem quite relevant. First, how does the accuracy of the trained MTurkers affect the accuracy of scalable recognition? Second, how can machine teaching algorithms be enhanced to improve MTurker accuracy?

3.2. How Important is Annotator Accuracy?

Since the training of MTurkers is not perfect, labels can be noisy. In general, the human-labeled dataset $\mathcal{D}^u$ is noisier than if labeled by experts. This begs the question of how accurate must the trained MTurkers be for machine teaching to be useful. To determine this, we perform a set of experiments with simulated "noisy MTurkers." Given an unlabeled dataset $\mathcal{D}^u$, for which the ground-truth labels Y are known to us but unavailable to the algorithms, we assign to each image a noisy label Y', according to a confusion matrix $\mathbf{M}$, where $m_{ij} = P(Y' = i|Y = j)$. More precisely, given ground truth label $y = j$, a class label y' is sampled from the distribution $[m_{1j}, ..., m_{Cj}]$.

The resulting noisy labeled dataset $\mathcal{D}^n$ is used to train a classifier f. By comparing the accuracy of f to that of a classifier g trained on the ground truth dataset, it is possible to determine the effect of MTurker annotation noise on the final classification performance. By varying the matrix $\mathbf{M}$, it is possible to analyze how the latter depends on the quality of the annotators. To enable these comparisons, we propose two metrics. The first is the **labeling accuracy**

$$\text{ACC}^l = \frac{\sum_{i=1}^{C} m_{ii}}{\sum_{i=1}^{C} \sum_{j=1}^{C} m_{ij}}. \tag{1}$$

This is a number in $[0, 1]$, equal to 1 when there is no labeling noise. The second is the **classification accuracy**, measured by average accuracy on the testing set $\mathcal{T}$ of classifiers f trained on $\mathcal{D}^l \cup \mathcal{D}^n$.

To investigate the effects of the confusion matrix $\mathbf{M}$ on classifier accuracy, we considered nine different matrices. The first five were estimated from real MTurker data. Annotators were trained with several machine teaching algorithms from the literature, chosen to reflect the spectrum of training effectiveness. The weakest performance was implemented with the RANDOM [28, 40] procedure, where annotators are taught with a randomly chosen teaching set $\mathcal{L}$. Stronger performances were implemented with omniIMT [25], imiIMT [25], bbIMT [26] as well as the state-of-the-art MaxGrad machine teaching algorithm [40]. As can be seen in Figure 2, the latter four produce much more accurate annotators than the former. The next four matrices are hand-crafted models of annotator quality. The first is a "chance level" annotator, i.e. $m_{ij} = 1/C, \forall i, j$. The next two are models that mimic matrices estimated on MTurk. They are denoted as diag-60 and diag-80, and have diagonal elements of 0.6, 0.8, respectively, and uniform non-diagonal values. diag-60 approximates RANDOM and diag-80 approximates MaxGrad. The final model is a perfect annotator with a diagonal matrix $\mathbf{M}$ of entries 1.

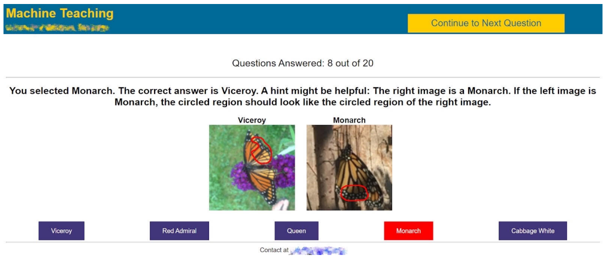
Figure 3 summarizes the result of this experiment, enabling several interesting observations. First, the labeling accuracies of diag-60 and diag-80 do match those of RANDOM and MaxGrad, respectively. However, the same does not hold for the associated classification accuracies. In fact, one of the most interesting observations of the figure is how the hand-crafted matrices have much weaker classification accuracy than those learned from MTurker data. In particular, the classification accuracy is always higher than the labeling accuracy for the MTurk matrices, but the reverse holds for their models.

A closer inspection of the confusion matrices shows that those estimated from human annotators do not have a uniform distribution for the annotation errors. While the diagonal value may not be 1, there is usually a dominant class for mistakes, i.e. the second probability tends to be larger than the remaining. This is likely to simplify the learning of the classifier, since it is mostly faced with label noise between pairs of classes, rather than all. The ensuing insight is that, beyond errors, it also matters what type of errors are made by the annotators. Informative labeling errors, between a few classes, lead to much better classifiers than uninformative, uniformly distributed, ones. Note that the differences in classification accuracy are substantial, with the MTurk-trained classifiers outperforming the model-trained classifiers by $5-10\%$.

Having said this, a second observation is that the *accuracy of the machine teaching algorithm does matter.* For example, both MaxGrad and diag-80 produced better classifiers than RANDOM and all methods produced very large gains over the chance annotator. Comparing machine teaching algorithms, it is clear that recognition accuracy increases with labeling accuracy. Finally, it can be observed that there is an upper bound on the required annotator accuracy. In fact, the perfect annotator produces classifiers that are only marginally better than those of MaxGrad. This is quite interesting, suggesting that current machine teaching algorithms already are a viable solution for classifier training. We note, however, that this is an experiment based on five classes. For large C, the differences are likely to be more significant. This is left for future research.

3.3. The Role of Explanations

A machine teaching algorithm aims to select the teaching set $\mathcal{L}$ from $\mathcal{D}^l$ that maximizes student labeling accuracy. Traditional algorithms [34, 25] present the images in $\mathcal{L}$ to the student, displaying the ground truth label as feedback when the latter makes a mistake. While this can suffice for coarse-grained classification, it is not ideal for most expert domains, where classification tends to be fine-grained. In this case, the differences between categories can be imperceptible to the untrained eye. Without further hints, it can be quite hard for non-experts to learn the target concepts. [28] addressed the problem with the EXPLAIN algorithm, which introduced attributive explanations into machine teaching. These are explanations based on a saliency map that highlights regions contributing to the classifier prediction [33, 46]. By directing student attention to features important for the classification, these explanations can enhance teaching. However, more recent methods, such as bbIMT [26], imiIMT [25], or MaxGrad [40] achieve better results than EXPLAIN without explanations.



Questions Answered: 8 out of 20

You selected Monarch. The correct answer is Viceroy. A hint might be helpful: The right image is a Monarch. If the left image is Monarch, the circled region should look like the circled region of the right image.

Figure 4: Interface. When the teaching image is “Viceroy” but the worker selected “Monarch”, the shown feedback will be given.

In this work, we seek to add explanations to the state-of-the-art MaxGrad algorithm [40]. We note, however, that a limitation of attributive explanations, such as those of EXPLAIN, is the lack of user-specific interaction. At each teaching iteration, the feedback provided by these explanations is always the correct label and the corresponding attribution map. Since the class predicted by the student is not considered in the explanation, the latter does not necessarily address the student's difficulties. Better feedback should take the student prediction into account. This is the definition of counterfactual explanations [41, 8], which address the question: “why is the class predicted by the student incorrect?” We next introduce an enhanced version of MaxGrad that leverages counterfactual explanations.

3.4. Counterfactual MaxGrad (CMaxGrad)

MaxGrad uses the iterative teaching strategy popular in the literature [25, 26, 40]. A network (f^1) initialized with an ImageNet pre-trained model is used to model the student. The MaxGrad teacher builds the teaching set iteratively, by extracting from $\mathcal{D}^l$ the images most informative for the student. In particular, at iteration t, the teacher selects an image $\mathbf{x}^t$ from $\mathcal{D}^l - \mathcal{L}^{t-1}$, where $\mathcal{L}^{t-1}$ is the teaching set assembled at iteration $t-1$. The teaching set is then augmented into $\mathcal{L}^t = \mathcal{L}^{t-1} \cup \{\mathbf{x}^t\}$ and used to retrain the student into $f^t = f^*(\mathcal{L}^t)$, where f^* denotes optimal classifier. The complete algorithm is given in Algorithm 1.

Counterfactual explanations can provide detailed student feedback during the retraining step when, given the query image $\mathbf{x}^t$ of ground-truth label y^t, the student predicts a counterfactual class $y^c \neq y^t$. An example is shown in Fig-

Algorithm 1 MaxGrad

```
Input Data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{M}$, max iter. $T$.
Initialization: $\mathcal{L}^0 \leftarrow \emptyset, f^1, \mathcal{D}^0 \leftarrow \mathcal{D}$
for $t = \{1, \dots, T\}$ do
    compute $\xi(\mathbf{x}_i)$ for all examples in $\mathcal{D}^{t-1}$
    select $\mathbf{x}^t = \arg\max_{\mathbf{x}_i \in \mathcal{D}^{t-1}} \xi(\mathbf{x}_i)$
    teaching set update: $\mathcal{L}^t \leftarrow \mathcal{L}^{t-1} \cup \{\mathbf{x}^t\}$
    student update: $f^{t+1} = f^*(\mathcal{L}^t)$
    $\mathcal{D}^t \leftarrow \mathcal{D}^{t-1} \setminus \{\mathbf{x}^t\}$
end for
Output $\mathcal{L}^t$
```

Algorithm 2 CMaxGrad

```
Input Data $\mathcal{D}^l = \{(\mathbf{x}_i, y_i)\}_{i=1}^{M}$, max iter. $T$, $\alpha$ and $\beta$, $\mathcal{E} = \{\mathbf{c}^{y^c}(\mathbf{x}_i) | y^c \neq y_i\}_{i=1,c=1}^{M,C}$.
Initialization: $\mathcal{L}^0 \leftarrow \emptyset, \mathcal{C}^0 \leftarrow \emptyset, f^1, \mathcal{D}^0 \leftarrow \mathcal{D}^l, \mathcal{E}^0 \leftarrow \mathcal{E}$
for $t = \{1, \dots, T\}$ do
    compute $\xi(\mathbf{x}_i)$ for all examples in $\mathcal{D}^{t-1}$ and $\xi(\mathbf{c}^{f^t(\mathbf{x}_i)}(\mathbf{x}_i))$ for all examples in $\mathcal{E}^{t-1}$
    select $\mathbf{x}^t = \arg\max_{\{\mathbf{x}_i \in \mathcal{D}^l - \mathcal{L}^{t-1}\}} \xi_c(\mathbf{x}_i, \mathbf{c}^{f^t(\mathbf{x}_i)}(\mathbf{x}_i); \alpha)$
    select $\mathbf{x}^{t,c} = \arg\max_{\{\mathbf{x}_i \in \mathcal{D}^l - \mathcal{L}^{t-1} | y_i = f^t(\mathbf{x}^t)\}} \xi_c(\mathbf{x}_i, c^{y^t}(\mathbf{x}_i); \beta)$
    teaching and explanation sets update: $\mathcal{L}^t \leftarrow \mathcal{L}^{t-1} \cup \{\mathbf{x}^t\}$, $\mathcal{C}^t \leftarrow \mathcal{C}^{t-1} \cup \{\mathbf{x}^{t,c}, \mathbf{c}^{f^t(\mathbf{x}^t)}(\mathbf{x}^t), \mathbf{c}^{y^t}(\mathbf{x}^{t,c})\}$
    student update: $f^{t+1} = f^*(\mathcal{L}^t \cup \mathcal{C}^t)$
    $\mathcal{D}^t \leftarrow \mathcal{D}^{t-1} \setminus \{\mathbf{x}^t, \mathbf{x}^{t,c}\}$, $\mathcal{E}^t \leftarrow \mathcal{E}^{t-1} \setminus \{\mathbf{c}^{f^t(\mathbf{x}^t)}(\mathbf{x}^t), \mathbf{c}^{y^t}(\mathbf{x}^{t,c})\}$
end for
Output $\mathcal{L}^t$
```

ure 4 for the Butterflies dataset, where y^t = 'Viceroy' and y^c = 'Monarch'. The explanation first samples an image $\mathbf{x}^c$ from y^c, and then produces a visualization of the form: "The correct label is y^t. If the correct label were y^c, the circled region of $\mathbf{x}^t$ should look like the circled region of x^c." Mathematically, this reduces to a function

$$\mathcal{C}(\mathbf{x}^t, y^t, y^c, \mathbf{x}^c) = (\mathbf{c}^c(\mathbf{x}^t), \mathbf{c}^t(\mathbf{x}^c)), \tag{2}$$

where $\mathbf{c}^c(\mathbf{x}^t)$ and $\mathbf{c}^t(\mathbf{x}^c)$ are counterfactual heatmaps or segments for images $\mathbf{x}^t$ and $\mathbf{x}^c$ respectively. They highlight image regions of features discriminant for the two classes. In Figure 4, these are the presence/absence of a line that crosses the radial wing lines of the two butterflies, and the different configurations of white spots. This explanation allows the student to quickly learn what to look for in order to distinguish the two classes. Since the counterfactual class was selected by the student, the process quickly provides the student with *precise* feedback on how to differentiate between the classes that most *confuse* them.

To include counterfactual explanations on MaxGrad, we propose the following generalization.

1. counterfactual maps are generated for all pairs of queries and counterfactual examples in the labeled dataset $\mathcal{D}^l$. This results in the explanation set $\mathcal{E} = \{\mathbf{c}^{y^c}(\mathbf{x}_i) | y^c \neq y_i\}_{i=1,c=1}^{M,C}$. This is a pre-processing step, performed before machine teaching takes place.

2. teaching set $\mathcal{L}^t$ is augmented with a *counterfactual set* $\mathcal{C}^t$ that includes counterfactual images and heatmaps.

3. during training, at iteration t the teacher selects an image $\mathbf{x}^t$ from $\mathcal{D}^l - \mathcal{L}^{t-1}$. The student then makes a prediction $y = f^t(\mathbf{x}^t)$. For the reasons discussed below, this is always incorrect i.e. $y = y^c \neq y^t$, A counterfactual image $\mathbf{x}^{t,c}$ is selected from class y^c and the counterfactual maps $(\mathbf{c}^c(\mathbf{x}^t), \mathbf{c}^t(\mathbf{x}^{t,c}))$ are retrieved from $\mathcal{E}$. The teaching set is then augmented into $\mathcal{L}^t = \mathcal{L}^{t-1} \cup \{\mathbf{x}^t\}$ and the counterfactual set into $\mathcal{C}^t = \mathcal{C}^{t-1} \cup \{\mathbf{x}^{t,c}, \mathbf{c}^c(\mathbf{x}^t), \mathbf{c}^t(\mathbf{x}^{t,c})\}$. The student is finally updated with $f^{t+1} = f^*(\mathcal{L}^t \cup \mathcal{C}^t)$.

In MaxGrad, the image $\mathbf{x}^t$ selected by the teacher is the one that maximizes a score $\xi(\mathbf{x})$ representative of the classification difficulty posed by image $\mathbf{x}$ to the student model f^t. Since this score is the negative classification margin $\xi(\mathbf{x})$ of the image $\mathbf{x}$ under f^t, there is always at least one image that the student cannot classify correctly in $\mathcal{D}^l - \mathcal{L}^{t-1}$ (otherwise the training would be complete). Hence, the resulting student prediction is incorrect, i.e. a counterfactual class $y^{t,c}$.

However, in the counterfactual setting, image selection must also account for the counterfactual heatmaps $(\mathbf{c}^c(\mathbf{x}^t), \mathbf{c}^t(\mathbf{x}^{t,c}))$. For this, we propose a *counterfactual margin score*

$$\xi_c(\mathbf{x}, \mathbf{c}^y(\mathbf{x}); \alpha) = \alpha\xi(\mathbf{x}) + (1-\alpha)\xi(\mathbf{c}^y(\mathbf{x})), \tag{3}$$

where $\alpha \in [0, 1]$ is a hyperparameter that weighs the contribution of images and counterfactual regions. Note that this supports scores based on the margin of the whole image ($\alpha = 1$), the counterfactual region ($\alpha = 0$) or both. This leads to the following procedure for the selection of the image $\mathbf{x}^t$ to augment the teaching set. For each image $\mathbf{x}_i \in \mathcal{D}^l - \mathcal{L}^{t-1}$, the counterfactual class is identified as $f^t(\mathbf{x}_i)$ and the heatmap $\mathbf{c}^{f^t(\mathbf{x}_i)}(\mathbf{x}_i)$ retrieved from $\mathcal{E}$. The teacher then selects the image of largest score, i.e.

$$\mathbf{x}^t = \arg\max_{\{\mathbf{x}_i \in \mathcal{D}^l - \mathcal{L}^{t-1}\}} \xi_c(\mathbf{x}_i, \mathbf{c}^{f^t(\mathbf{x}_i)}(\mathbf{x}_i); \alpha), \tag{4}$$

to add to the teaching set $\mathcal{L}^{t-1}$.

The image $\mathbf{x}^{t,c}$ of the counterfactual class $y^{t,c}$ is then chosen with the same criterion among the images in the counterfactual class, i.e.

$$\mathbf{x}^{t,c} = \arg\max_{\{\mathbf{x}_i \in \mathcal{D}^l - \mathcal{L}^{t-1} | y_i = f^t(\mathbf{x}^t)\}} \xi_c(\mathbf{x}_i, c^{y^t}(\mathbf{x}_i); \beta), \tag{5}$$

	Butterflies	Chinese Char.
RANDOM	65.20	47.05
STRICT [34]	65.00	51.51
EXPLAIN [28]	68.33	65.44
omniIMT [25]	70.07 (18.30)	64.36 (19.58)
imiIMT [25]	72.70 (17.63)	64.46 (23.72)
bbIMT [26]	76.09 (18.05)	64.37 (19.57)
MaxGrad [40]	80.33 (19.76)	81.89 (12.93)
CMaxGrad	**84.10** (18.24)	**84.63** (20.18)

Table 1: Test set labeling accuracy, mean (std), of MTurkers.

where y^t is the label of $\mathbf{x}^t$. The teaching set $\mathcal{L}^t$ and the counterfactual set $\mathcal{C}^t$ are then updated with $\mathbf{x}^t$ and $(\mathbf{x}^{t,c}, \mathbf{c}^{f^t(\mathbf{x}^t)}(\mathbf{x}^t), \mathbf{c}^{y^t}(\mathbf{x}^{t,c}))$, respectively, and the student updated with $f^{t+1} = f^*(\mathcal{L}^t \cup \mathcal{C}^t)$. This requires training a classifier with both images and image regions, derived from the counterfactual heatmaps. In our implementation, counterfactual regions are converted to images by simply thresholding the heatmaps and setting the pixels outside the counterfactual region to the average image color. The resulting images are then added to $\mathcal{C}^t$. We note, however, that this is not done on human teaching experiments, where subjects are shown whole images, as demonstrated in Figure 4. The overall procedure is summarized in Algorithm 2 and denoted CMaxGrad.

4. Evaluation of Student Teaching

We start by evaluating the accuracy of the labels produced by students trained with CMaxGrad. Following [40, 28], we consider both simulated and real students.

Dataset We used two recent machine teaching benchmark datasets: Butterflies and Chinese Characters [28]. These are more challenging than binary classification or synthetic datasets used in earlier work [34, 5, 36], because they are both fine-grained multi-class datasets of real images from expert domains. Butterflies has five butterfly species sampled from iNaturalist [38], with 1544 training and 386 testing samples. Chinese Characters consists of three similar Chinese characters, with 568 training and 143 testing examples. Both datasets have large intra-class diversity, e.g. due to different handwriting styles, and large inter-class similarity.

Network Counterfactual explanations are generated by a ResNet-18 pre-trained in ImageNet and fine-tuned on the target training set. The student is simulated with a ResNet-18 pre-trained on ImageNet. This mimics a student that starts from a good generic understanding of image classification but has no expertise in the target domain.

Teaching All experiments use a teaching set of 20 examples, selected from the training set and tested on the testing set. Counterfactual maps were generated with the recent SCOUT algorithm [41]. Counterfactual regions were extracted by setting the segment size parameter to 5% of the image area. The parameters α, β of (4) and (5), respectively, were set to $\alpha = \beta = 0.5$ after cross-validation. In the supplement we also present results for experiments with simulated students, which enable replicable method comparisons, and a detailed description of the set-up used to train MTurkers, using the interface of Figure 4. Table 1 reports the test accuracies of workers trained with different methods from the literature. Counterfactual explanations enabled a significant improvement in the accuracy of the MTurk student labels.

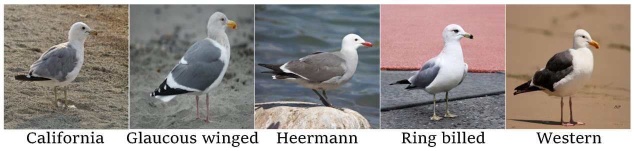


Figure 5: Sample images of Gull dataset.

5. Evaluation of Scalable Recognition

In this section, we evaluate the performance of the complete architecture of Figure 1.

Dataset Because there is no benchmark for the evaluation of scalable fine-grained recognition in expert domains, we created two such benchmarks. The first is based on the Butterflies dataset. The first 300 training samples (according to the dataset order[1]) compose the expert labeled dataset $\mathcal{D}^l$, and the remaining 1,244 the unlabeled dataset $\mathcal{D}^u$ to be annotated by Mturkers. The testing set is used for evaluation. The second benchmark is from an even more fine-grained and thus difficult task, based on the recognition of five gull categories: "California Gull", "Glaucous winged Gull", "Heermann Gull", "Ring billed Gull" and "Western Gull". An example image from each class is shown in Figure 5. These classes were chosen because they are the overlapping classes of two widely used bird datasets, CUB200 [42] and NAbirds [37]. The images from the CUB training set (150 instances) serve as expert-labeled dataset $\mathcal{D}^l$ whereas those from NAbirds serve as unlabeled dataset $\mathcal{D}^u$ (431 instances). The CUB testing set (149 instances) is used for evaluation.

Network A ResNet-18 is used as classifier. Explanations are generated by two models, each specific to one dataset. Because two of the butterfly categories are in ImageNet, the ResNet-18 is initialized from scratch for the Butterflies dataset. The Gull dataset has no overlap with ImageNet and is more challenging. Since the network trained from scratch on this dataset performs only slightly better than chance level ($\approx 30\%$), the network is initialized with the model pre-trained on ImageNet.

Platform All experiments were conducted on Amazon Mechanical Turk. Each MTurker received a teaching set of 20

[1] https://github.com/macaodha/explain_teach

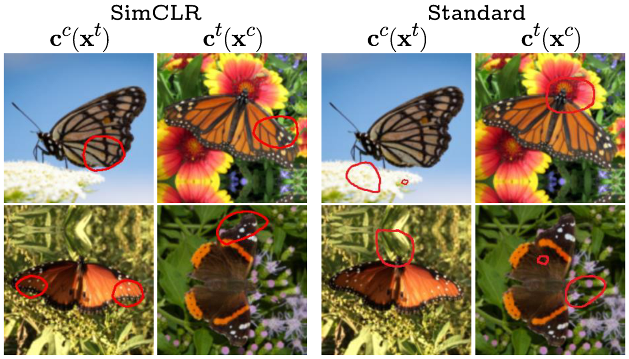

SimCLR
Standard
$\mathbf{c}^c(\mathbf{x}^t)$
$\mathbf{c}^t(\mathbf{x}^c)$
$\mathbf{c}^c(\mathbf{x}^t)$
$\mathbf{c}^t(\mathbf{x}^c)$

Figure 6: Comparison of counterfactual explanations generated by different models. Two examples are shown. Top: true class is “Viceroy” and counter class is “Monarch”; bottom: true class is “Queen” and counter class is “Red Admiral”.

examples, chosen by CMaxGrad, and was then requested to label 30 images randomly sampled from the unlabeled set. This number was chosen so as to avoid the danger of worker fatigue and frustration possible with larger jobs. Three labelings were collected per example and their majority vote was chosen as the final label. If the labels were distinct, we chose one randomly.

Baselines MEMORABLE was compared to a number of scalable recognition baselines, whose results are shown in the top part of Table 2. “Supervised” refers to vanilla supervised learning on the expert-labeled dataset $\mathcal{D}^l$. Pseudo-Label [21] first trains with supervision on $\mathcal{D}^l$, then iteratively improves the performance by self-labeling the unlabeled examples in $\mathcal{D}^u$ and training on the pseudo labels. SimCLR [3] is a representative semi-supervised learning method.

Results The second part of Table 2 presents results of MEMORABLE, using two strategies to train the classifier that produces the counterfactual explanations. The first is supervised learning on the expert-labeled dataset $\mathcal{D}^l$. The second is semi-supervised learning on $\mathcal{D}^l$ and $\mathcal{D}^u$. For the latter, we adopted the SimCLR [3] contrastive learning algorithm. This is denoted with ”+SimCLR”in Table 2.

When compared to MaxGrad, the counterfactual explanations produced by CMaxGrad enable substantial better classification accuracies, e.g. a gain of about 4% on Gull. For CMaxGrad, best performance was achieved with the semi-supervised model, with which MEMORABLE outperformed the best baseline by 1.3% on Butterflies but without the semi-supervised model by 1.5% on the more challenging Gull dataset. This is consistent with the performance of the classifier. Figure 6 shows examples of counterfactual regions selected by the two versions of CMaxGrad. While those produce with SimCLR cover body parts, the supervised model sometimes has difficulty localizing the class-discriminant regions, perhaps due to its lower classification accuracy.

	Butterflies	Gull
Supervised baseline	59.4 (1.3)	58.3 (0.6)
Pseudo-Label [21]	64.7 (1.1)	58.7 (1.4)
SimCLR [3]	76.9 (0.4)	53.0 (0.8)
MaxGrad	74.7 (0.7)	56.3 (0.9)
CMaxGrad	77.5 (0.5)	60.2 (1.1)
CMaxGrad+SimCLR	78.2 (0.2)	59.7 (0.7)
MaxGrad+DivideMix	78.6 (1.2)	59.9 (1.2)
CMaxGrad+DivideMix	81.2 (1.1)	61.7 (1.5)
CMaxGrad+SimCLR+DivideMix	83.4 (0.7)	61.2 (1.1)

Table 2: Test accuracy comparison with mean (std). The lower group shows our results whereas the upper other literature.

Enhancements Since the labels produced by MTurkers are noisy, further performance improvements can in principle be accrued by training the final classifier with noisy label learning algorithms [22, 9, 35]. The bottom part of Table 2 shows results obtained with the state of the art DivideMix method [22]. Somewhat surprisingly, DivideMix was always able to improve results significantly. Note that even the combination CMaxGrad+DivideMix outperformed the best baseline by $3-5\%$ on these datasets. When further combined with SimCLR-based explanations, the gains were of about 6% on Butterflies. This suggests that even when the MTurker labels are incorrect they are informative of the true class, as discussed in Section 3.2. It also shows that MEMORABLE is a viable alternative to scalable recognition, especially in expert domains.

6. Conclusion

In this paper, we proposed the MEMORABLE framework for scalable recognition in fine-grained expert domains. This is based on the novel CMaxGrad machine teaching algorithm, which leverages counterfactual explanations to account for student predictions during the teaching process. We have also conducted the first studies of machine teaching in the context of the entire scalable recognition pipeline. It was shown that both CMaxGrad and MEMORABLE achieve superior results to existing solutions to their respective problems. It could be argued that comparing MEMORABLE to previous scalable recognition methods is unfair, since it leverages additional resources in the form of crowdsourcing. While this is true, we argue that crowdsourcing platforms are now very accessible and dataset labeling is a one-time cost. This must be weighed against the benefits of a better dataset that, as shown by the recent computer vision history, is a gift that keeps on giving.

Acknowledgement This work was partially funded by NSF awards IIS-1924937, IIS-2041009, a gift from Amazon, a gift from Qualcomm, and NVIDIA GPU donations. We also acknowledge and thank the use of the Nautilus platform for some of the experiments discussed above.

References

[1] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Grounding visual explanations. In *ECCV*, 2018. 3

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 3

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2, 3, 8

[4] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 3

[5] Yuxin Chen, Oisin Mac Aodha, Shihan Su, Pietro Perona, and Yisong Yue. Near-optimal machine teaching via explanatory teaching sets. In *International Conference on Artificial Intelligence and Statistics*, pages 1970–1978. PMLR, 2018. 3, 7

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[7] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *NeurIPS*, 2018. 3

[8] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. *ICML*, 2019. 2, 3, 5

[9] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *arXiv preprint arXiv:1804.06872*, 2018. 2, 8

[10] Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Generating counterfactual explanations with natural language. *arXiv preprint arXiv:1806.09809*, 2018. 3

[11] http://spc.ucsd.edu/. 3

[12] https://www.clickworker.com/. 2

[13] https://www.microworkers.com/. 2

[14] https://www.mturk.com/. 1, 2

[15] https://www.sama.com/. 2

[16] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2021. 3

[17] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 3

[18] Edward Johns, Oisin Mac Aodha, and Gabriel J Brostow. Becoming the expert-interactive multi-class machine teaching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2624, 2015. 2, 3

[19] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1

[20] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Shahab Kamali, Matteo Malloci, Jordi Pont-Tuset, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017. 1

[21] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 2013. 8

[22] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *ICLR*, 2020. 2, 8

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[24] Shusen Liu, Bhavya Kailkhura, Donald Loveland, and Yong Han. Generative counterfactual introspection for explainable deep learning. *arXiv preprint arXiv:1907.03077*, 2019. 3

[25] Weiyang Liu, Bo Dai, Ahmad Humayun, Charlene Tay, Chen Yu, Linda B Smith, James M Rehg, and Le Song. Iterative machine teaching. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2149–2158. JMLR. org, 2017. 2, 4, 5, 7

[26] Weiyang Liu, Bo Dai, Xingguo Li, Zhen Liu, James M Rehg, and Le Song. Towards black-box iterative machine teaching. *International Conference on Machine Learning*, 2018. 2, 4, 5, 7

[27] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *arXiv preprint arXiv:2006.08218*, 1(2), 2020. 3

[28] Oisin Mac Aodha, Shihan Su, Yuxin Chen, Pietro Perona, and Yisong Yue. Teaching categories to human learners with visual explanations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3820–3828, 2018. 2, 3, 4, 5, 7

[29] Mohammad Sadegh Norouzzadeh, Anh Nguyen, Margaret Kosmala, Alexandra Swanson, Meredith S Palmer, Craig Packer, and Jeff Clune. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*, 115(25):E5716–E5725, 2018. 3

[30] Kaustubh R Patil, Xiaojin Zhu, Łukasz Kopeć, and Bradley C Love. Optimal teaching for limited-capacity human learners. In *Advances in neural information processing systems*, pages 2465–2473, 2014. 3

[31] Genevieve Patterson, Grant Van Horn, Serge Belongie, Pietro Perona, and James Hays. Tropel: Crowdsourcing detectors with minimal training. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 3, 2015. 3

[32] Gordon B Schmidt and William M Jettinghoff. Using amazon mechanical turk and other compensated crowdsourcing sites. *Business Horizons*, 59(4):391–400, 2016. 2

[33] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 2, 5

[34] Adish Singla, Ilija Bogunovic, Gábor Bartók, Amin Karbasi, and Andreas Krause. Near-optimally teaching the crowd to classify. In *International Conference on Machine Learning*, pages 154–162. PMLR, 2014. 3, 4, 5, 7

[35] Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *arXiv preprint arXiv:2007.08199*, 2020. 8

[36] Shihan Su, Yuxin Chen, Oisin Mac Aodha, Pietro Perona, and Yisong Yue. Interpretable machine teaching via feature feedback. 2017. 3, 7

[37] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 595–604, 2015. 1, 3, 7

[38] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 1, 7

[39] Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. *arXiv preprint arXiv:1907.02584*, 2019. 3

[40] Pei Wang, Kabir Nagrecha, and Nuno Vasconcelos. Gradient-based algorithms for machine teaching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1387–1396, June 2021. 2, 3, 4, 5, 7

[41] Pei Wang and Nuno Vasconcelos. Scout: Self-aware discriminant counterfactual explanations. *CVPR*, 2020. 2, 3, 5, 7

[42] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 7

[43] Tete Xiao, Xiaolong Wang, Alexei A Efros, and Trevor Darrell. What should not be contrastive in contrastive learning. *arXiv preprint arXiv:2008.05659*, 2020. 2

[44] Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *arXiv preprint arXiv:2103.00550*, 2021. 3, 4

[45] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1476–1485, 2019. 3, 4

[46] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 2, 3, 5

[47] Xiaojin Zhu. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015. 2, 3

[48] Xiaojin Zhu, Adish Singla, Sandra Zilles, and Anna N Rafferty. An overview of machine teaching. *arXiv preprint arXiv:1801.05927*, 2018. 2, 3

[49] Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. *ICLR*, 2017. 3